

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу  
Кафедра математичних методів системного аналізу**

«На правах рукопису»  
УДК 681.3

«До захисту допущено»  
Завідувач кафедри  
\_\_\_\_\_ О.Л. Тимошук  
«\_\_\_» \_\_\_\_\_ 2018 р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

**зі спеціальності 124 Системний аналіз**

**на тему: «Побудова системи підтримки прийняття рішень для  
підприємств роздрібної торгівлі на основі методів інтелектуального  
аналізу даних»**

Виконав:  
студент II курсу, групи КА-62м  
Тарасюк Тарас Сергійович \_\_\_\_\_

Керівник:  
професор, д.т.н., проф.  
Данилов В.Я. \_\_\_\_\_

Рецензент:  
Каф. ІБ, професор НТУУ «КПІ»  
імені Сікорського, д.т.н.  
Качинський А.Б. \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

**Інститут прикладного системного аналізу**

**Кафедра математичних методів системного аналізу**

Рівень вищої освіти – другий (магістерський)

Спеціальність (спеціалізація) – 124 Системний аналіз (Системи і методи підтримки прийняття рішень)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

\_\_\_\_\_ О.Л. Тимошук

«\_\_» \_\_\_\_\_ 2018 р.

**ЗАВДАННЯ**

**на магістерську дисертацію студенту**

**Тарасюку Тарасу Сергійовичу**

1. Тема дисертації «Побудова системи підтримки прийняття рішень для підприємств роздрібної торгівлі на основі методів інтелектуального аналізу даних», науковий керівник дисертації Данилов Валерій Якович, д.т.н, проф., затверджені наказом по університету від «\_\_» \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін подання студентом дисертації \_\_\_\_\_

3. Об'єкт дослідження: статистичні ряди даних, що описують бізнес-процеси роздрібної торгівлі.

4. Предмет дослідження: моделі та методи інтелектуального аналізу даних.

5. Перелік завдань, які потрібно розробити: огляд існуючих проблем роздрібної торгівлі, огляд існуючих моделей та методів, збір та підготовка даних, побудова та реалізація моделей та методів, проведення експериментів, аналіз отриманих результатів.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: постановка завдання дослідження, підготовлені дані, побудова моделей, порівняння результатів.

7. Орієнтовний перелік публікацій: «Методи інтелектуального аналізу даних в роздрібній торгівлі» - стаття в міжнародному електронному науковому журналі «Наука онлайн», «Система підтримки прийняття рішень у роздрібній торгівлі» - тези міжнародної наукової конференції «Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту (ISDMCI'2018)»

8. \_\_\_\_\_ Дата \_\_\_\_\_ видачі \_\_\_\_\_ завдання \_\_\_\_\_

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка

Студент

Т.С. Тарасюк

Науковий керівник дисертації

В.Я. Данилов

## РЕФЕРАТ

Магістерська дисертація: 132 с., 31 рис., 33 табл., 2 додатки і 27 джерел.

Актуальність теми: у зв'язку зі зниженням питомої вартості зберігання даних, зростає обсяг інформації, що накопичують підприємства роздрібної торгівлі, а це спричиняє виникнення завдань, пов'язаних з необхідністю обробки великих масивів даних з метою пошуку нових латентних закономірностей, встановлення і виявлення нових знань. Тому в магістерській дисертації здійснюється проектування та реалізація системи підтримки прийняття рішень, що забезпечує пошук асоціацій, моделювання та прогнозування ключових фінансових показників підприємства та сприяє створенню стратегій розвитку підприємства.

Мета дослідження – розробка структури СППР та її реалізація у вигляді програмного продукту для побудови асоціацій, моделювання та прогнозування бізнес-процесів у роздрібній торгівлі.

Об'єкт дослідження – статистичні ряди даних бізнес-процесів роздрібної торгівлі.

Предмет дослідження – моделі та методи інтелектуального аналізу даних: алгоритм Apriori для генерації асоціативних правил, моделі регресійного аналізу з прогнозування часових рядів: AR, ARMA, ARIMA та моделей у вигляді тренду.

Методи дослідження – методи асоціативного аналізу та регресійного аналізу.

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, РОЗДРІБНА ТОРГІВЛЯ,  
АСОЦІАТИВНИЙ АНАЛІЗ, РЕГРЕСІЙНИЙ АНАЛІЗ, СИСТЕМА  
ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

## ABSTRACT

Master's thesis: 132 p., 31 f., 33 tables, 2 appendix, 27 sources.

Subject relevance - in connection with the decrease in the specific value of data storage, the amount of information accumulated by retailers is increasing, this entails the emergence of tasks related to the need to process large amounts of data in order to find new patterns, the establishment and discovery of new knowledge. Therefore, in the master's thesis, the design and implementation of decision-support systems are provided, which ensures the search for associations, modeling and forecasting of key financial indicators of the retail enterprise and promotes the establishment of business strategies.

Purpose of research is to develop a decision support system structure and implement it as a new software product for building associations, modeling and forecasting business processes in retail.

Object of research - is the statistical series of data describing the business processes of the retail industry in need of efficient analytical processing to reveal practically useful knowledge about their interactions, necessary in the retail decision-making process.

Subject of research - models and methods of data mining: Apriori algorithm for generating associative rules, regression analysis model for forecasting of time series: AR, ARMA, ARIMA and trend models.

Research methods are based the theory of associative analysis and time series theory.

DATA MINING, RETAIL, ASSOCIATIVE ANALYSIS, REGRESSION ANALYSIS, DECISION SUPPORT SYSTEM

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1 ХАРАКТЕРИСТИКА РОЗДРІБНОЇ ТОРГІВЛІ.....	9
1.1 Маркетингові рішення в роздрібній торгівлі .....	9
1.2 Інструменти управління асортиментом товарів .....	11
1.3 Актуальність проблеми і постановка задачі .....	17
Висновки до розділу .....	19
РОЗДІЛ 2 ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ У РОЗДРІБНІЙ ТОРГІВЛІ .....	21
2.1 Побудова асоціативних правил .....	21
2.1.1 Синтез асоціативних правил .....	21
2.1.2 Аналіз видів асоціативних правил .....	23
2.1.3 Алгоритм Apriori .....	33
2.2 Прогнозування у роздрібній торгівлі .....	35
2.2.1 Моделі регресійного аналізу .....	36
2.2.2 Критерії адекватності моделі .....	39
Висновки до розділу .....	43
РОЗДІЛ 3 РОЗРОБКА ТА ЗАСТОСУВАННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ПІДПРИЄМСТВА РОЗДРІБНОЇ ТОРГІВЛІ ..	44
3.1 Системи підтримки прийняття рішень .....	44
3.2 Структура даних роздрібного магазину .....	46
3.3 Структура СППР .....	47
3.4 Довідка користувача з експлуатації застосунку .....	49

3.4.1 Формат вхідних даних .....	49
3.4.2 Завантаження вхідних даних .....	51
3.4.3 Попередній аналіз завантажених даних .....	53
3.4.4 Вибір та налаштування параметрів для асоціативного аналізу .....	55
3.4.5 Вибір та налаштування параметрів для регресійного аналізу .....	57
3.5 ABC-аналіз категорій товарів .....	59
3.6 Побудова та аналіз асоціативних правил .....	61
3.7 Побудова моделей ключових показників роздрібного магазину	64
3.7.1 Опис вихідних даних .....	64
3.7.2 Опис та аналіз отриманих моделей .....	65
Висновки до розділу .....	75
РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ .....	76
4.1 Опис ідеї проекту .....	76
4.2 Технологічний аудит ідеї проекту.....	77
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	78
4.4 Розроблення ринкової стратегії проекту .....	83
4.5 Розроблення маркетингової програми стартап-проекту.....	87
Висновки до розділу .....	90
ВИСНОВКИ.....	91
ПЕРЕЛІК ПОСИЛАНЬ .....	92
ДОДАТОК А ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДОПОВІДІ .....	95
ДОДАТОК Б ЛІСТИНГ КОДУ .....	105

## ВСТУП

За останні десятиліття підприємства досягли значних успіхів у сфері роздрібної торгівлі. Це стало можливим завдяки оптимізації збору, збереження та обробки інформації про клієнтів та їх покупки. На даний момент існує безліч систем, що дозволяють автоматизувати ці процеси та провести первинний аналіз отриманих даних. Але для подальшого розвитку підприємств роздрібної торгівлі та збільшення їхніх прибутків, необхідне забезпечення нових підходів, методів та прийомів аналізу зібраної інформації. Тому виникає потреба створення нових аналітичних систем, що базуючись на методах інтелектуального аналізу даних, зможуть надавати користувачу обґрунтовану інформацію для прийняття рішень про асортимент, розміщення та комплектацію товарів.

Інтелектуальний аналіз даних включає в себе методи класифікації, кластеризації, регресії та асоціативний аналіз, а також аналіз і прогнозування часових рядів. Вищезгаданий інструментарій дозволяє аналізувати, як інформацію про покупки та товари, так і інформацію про клієнтів. Класифікувати їх, розділяти на групи, знаходити залежності між покупками та враховувати часові аспекти.

Базуючись на результатах застосування різних моделей, методів та прийомів, необхідно забезпечити комплексний підхід для агрегації отриманих результатів, що дасть змогу аналітикам та користувачам робити конкретні висновки про доцільність продажу, розміщення товарів, їх комбінацій і використання системних знижок та допоможе при побудові стратегії розвитку підприємства.



## РОЗДІЛ 1 ХАРАКТЕРИСТИКА РОЗДРІБНОЇ ТОРГІВЛІ

### 1.1 Маркетингові рішення в роздрібній торгівлі

При здійсненні маркетингової діяльності в роздрібній торгівлі важливе значення має прийняття раціональних рішень, що являють собою вибір альтернатив із множини можливих варіантів здійснення діяльності, націлених на стійке формування, розвиток і задоволення попиту на товари чи послуги, а також систему заходів впливу на маркетингову систему для покращення чи збереження конкурентної переваги фірми, товару, для підтримки чи росту прибутку [1].

Рішення поділяються на стратегічні і тактичні. Стратегічні рішення присвячені формуванню місії і цілей підприємства роздрібної торгівлі, обґрунтуванню стратегій. В свою чергу, тактичні маркетингові рішення охоплюють визначення параметрів комплексу маркетингу підприємства [2].

В літературі можна виділити наявність різних підходів до змісту маркетингових рішень. Так, науковці виділяють в складі рішень наступне:

- формування асортименту підприємства роздрібної торгівлі;
- цінове рішення в роздрібній торгівлі;
- рішення по організації просування в роздрібній торгівлі [3].

На думку деяких авторів, основними рішеннями, що приймаються керівництвом роздрібних магазинів є:

- вибір цільового ринку;
- формування товарного асортименту і комплексу послуг;
- вибір методів ціноутворення і стимулювання збуту;
- вибір розміщення магазину.

Також існує точка зору, відповідно до якої рішення в роздрібній торгівлі розглядаються набагато ширше і охоплюють практично всі напрями маркетингової діяльності:

а) стратегічні рішення, націлені на підвищення привабливості роздрібних магазинів для клієнтів, що включають наступні підзадачі:

- прийняття рішень з визначення сильних і слабких сторін торгового підприємства на основі аналізу можливостей і ризиків, закладених у зовнішньому середовищі;

- формування стратегічних і тактичних цілей на основі результату моніторингу зовнішнього середовища підприємства;

- обґрунтування підходу до формування стратегії;

б) рішення по вибору цільового ринку, що визначають подальші рішення відносно асортименту, особливостей магазину, політики рекламування і ціноутворення;

в) рішення в сфері розробки комплексу маркетингу [4].

Для забезпечення обґрунтованості управлінських рішень необхідно провести аналіз зібраної інформації на основі даних про минуле і зробити висновки, що допоможуть приймати оптимальні рішення у майбутньому.

Аналіз даних роздрібної торгівлі поділяється на описову, приписну та інтелектуальну аналітики. Найбільш простою в застосуванні є описова аналітика, яка відноситься до методів, що використовуються для опису та звітування про минуле. Зазвичай використовується для підсумовування продажів за регіонами та рівнями інвентаризації. Приписна аналітика - це сукупність прийомів, які дозволяють визначати найкращу стратегію розвитку на основі заданих вимог і з метою підвищення ефективності бізнесу. У роздрібному бізнесі ми можемо використовувати приписну аналітику, щоб встановити зниження цін, встановити рівні знижок, максимізувати доходи. Інтелектуальна аналітика або аналітика прогнозування відноситься до методів, що використовуються в статистичних моделях та емпіричних методах на основі старих даних, для створення емпіричних прогнозів про майбутнє або визначення впливу однієї змінної на іншу. Роздрібні торговці можуть використовувати аналітику прогнозування, щоб отримати зразки

даних та прогнозувати майбутні продажі, задоволення клієнтів та можливість вибору онлайн-покупки. Комплекс застосування вищенаведених аналітик, дозволить отримати знання, необхідні керівнику підприємства роздрібно́ї торгівлі для прийняття обґрунтованих рішень.

## 1.2 Інструменти управління асортиментом товарів

Зміна поведінки покупців і загострення конкуренції змушують шукати нові способи управління асортиментом товарів, які дозволять більш ефективно позиціонувати товарні пропозиції. При цьому слід підтримувати і нарощувати їх ринкову частку, скорочуючи витрати виробництва та обігу товарів, збільшуючи прибутковість і підвищуючи ступінь задоволення зростаючих потреб покупців. На думку науковців, робота з асортиментом в роздрібній торгівлі полягає в створенні такої сукупності окремих товарів, яка має особливі переваги з точки зору покупців [5]. З цього випливає, що управління асортиментом є найважливішим завданням комерційної діяльності, маркетингу і конкурентної боротьби. Значення пошуку нових методів особливо зросло в останні десятиліття з точки зору економічного зростання роздрібного підприємства.

Управління асортиментом товарів у роздрібній торгівлі розглядалося багатьма вітчизняними та закордонними авторами в різних аспектах. Більшість вчених вивчають цей процес окремо, не зв'язуючи його з діяльністю всього підприємства. Відсутність системного підходу не дозволяє комплексно підійти до організації асортименту товарів. У зв'язку з цим існує необхідність в розробці сучасної методики управління, що дозволяє підвищити загальну економічну ефективність роздрібних торгових підприємств і найбільш повно задовольнити запити споживачів.

В даний час використовується безліч інструментів роботи з номенклатурою товарів, під якими розуміється сукупність прийомів, застосовуваних у сфері управління асортиментом. За допомогою інструментів можна проаналізувати такі параметри, як рівень прибутку (правило Парето), внесок товару в результат роботи магазину (ABC-аналіз), стабільність продажів (XYZ-аналіз), статус кожного товару в асортиментній матриці (суміщений ABC і XYZ-аналіз), час існування продукту і ринку (матриця Ансоффа), частка ринку і швидкість обсягу продажів (матриця BCG), ринкова привабливість і ефективність асортименту (матриця GE), час перебування товару на ринку (концепція орієнтації на життєвий цикл товарів (ЖЦТ)).

Правило Парето, або принцип «80/20», виявлено у 1906 році і виходить з того, що за 20% наслідків відповідає 80% причин. Згідно з твердженнями Вільфредо Парето, в будь-якому процесі мале число причин (20%) життєво важливо, а 80% не робить істотного впливу на результат [6].

Аспекти застосування цього інструменту для цілей управління асортиментом досить різноманітні, і однозначного алгоритму не існує. Однак в загальному вигляді це правило звучить так: двадцять відсотків асортименту товарів приносить вісімдесят відсотків прибутку, і, навпаки, вісімдесят відсотків асортименту товарів приносить двадцять відсотків прибутку. На основі такого поділу товарної номенклатури приймаються рішення по концентрації уваги і підвищенню рівня підтримки групи товарів, що приносить 80% прибутку, і скорочення групи товарів, що приносить інші 20% прибутку компанії.

Одним з універсальних і поширених інструментів роботи з асортиментом, що з'явився в 1948 році є ABC-аналіз, який ґрунтується на принципі Парето. В рамках загального рейтингового списку виділяють три групи об'єктів - А, В і С, які відрізняються за своєю значимістю і вкладу в обіг або прибутку магазину:

а) товари А – найважливіші товари, що приносять перші 80% прибутку або обороту;

б) товари В – середні за важливістю, що приносять ще 15% прибутку або обороту;

в) товари С – проблемні товари, що приносять інші 5% прибутку або обороту [7].

Таке співвідношення відсотків якраз і обумовлено правилом Парето: виділяються головні 80% прибутку або обороту і решту 20%. Очевидно, що необхідно жорстко контролювати наявність в асортименті товарних позицій класу «А». По відношенню до товарних позицій класу «В» контроль може бути поточним, а по відношенню до позицій класу «С» - періодичним.

XYZ-аналіз – статистичний інструмент, що дозволяє аналізувати і прогнозувати стабільність продажів окремих видів товарів і коливання рівня споживання тих чи інших ресурсів. Цей прийом зазвичай застосовується для ранжирування та групування асортиментних позицій за ступенем прогнозованості обсягу попиту або оборотності товару [8].

XYZ-аналізу схожий з ABC-аналізом і ґрунтується на тому ж принципі: товари поділяються на три групи – X, Y, Z, виходячи із значення коефіцієнта варіації за певний період часу. Цей аналіз ділить об'єкти за ступенем відхилення від середнього показника, вираховується за кілька періодів. XYZ-аналіз показує стабільність або нестабільність попиту. Чим стабільніший попит на товар, тим легше ним керувати, тим менша потреба в товарних запасах і тим легше планувати рух продукту.

Поєднаний ABC і XYZ-аналіз дозволяє більш точно розглянути асортимент і зрозуміти, як можна управляти тими чи іншими групами і категоріями товарів. В результаті даного поєднання виходить 9 груп об'єктів аналізу за двома критеріями - ступінь впливу на кінцевий результат (ABC) і стабільність (прогнозованість) цього результату (XYZ) [9].

Групи AX, AY і AZ вимагають найбільшої уваги, ретельного планування, обліку і контролю. Для ресурсів категорії CX, CY, CZ застосовуються укрупнені методи аналізу. Такий погляд на асортимент дає вже більш повну картину, і на підставі його можна приймати управлінські рішення більш обґрунтовано.

Матриця Ансоффа або матриця «товар-ринок» вперше була опублікована в 1957 р Ігорем Ансоффом, який запропонував розбиття асортименту на основі критеріїв часу присутності товару на ринку і часу існування відповідного ринку. Суть моделі полягає у виборі однієї з чотирьох базових стратегій досягнення цілей стратегічного розвитку компанії на ринку:

- а) проникнення на ринок - фірма виступає на існуючому ринку з існуючим товаром;
- б) розвиток ринку - фірма виступає на новому ринку, але з існуючим товаром;
- в) розробка товару - фірма виступає на існуючому ринку, але з новим товаром;
- г) диверсифікація - фірма виступає на новому ринку з новим товаром [10].

Матриця BCG заснована фахівцями Бостонської консалтингової групи в кінці 1960-х рр. Її класифікаційними ознаками є темпи зростання ринкового попиту, ставлення частки на ринку товару даної фірми до відповідної частки основного конкурента, обсяг інвестицій у виробництво товару, обсяг прибутків і ін. За результатами побудови матриці BCG виділяються чотири групи товарів (відповідно до попаданням конкретного товару в відповідний квадрант): «зірки», «дійні корови», «дикі кішки», «собаки». Для кожної з цих груп існує пріоритетна стратегія [11].

Матриця GE (McKinsey), що з'явилася на початку 1970-х рр., Спільно запропонована фірмою General Electric і консалтинговою компанією McKinsey & Co. В її основу лягли два показника: ринкова привабливість і

ефективність бізнесу (асортименту) [12]. Головною особливістю моделі є те, що в ній вперше для порівняння стали розглядати не тільки «фізичні» чинники (обсяг продажів, прибуток та інше), але і суб'єктивні характеристики, такі як мінливість частки ринку, технології, стан кадрового забезпечення.

Концепція орієнтації на життєвий цикл товару вперше була опублікована американським маркетингологом Теодором Левітом в 1965 р і виходить з того, що будь-який товар проходить певні стадії розвитку: стадію впровадження на ринок, стадію зростання, зрілості, старіння (спаду) [13]. Необхідність використання цієї концепції полягає в тому, що при знаходженні на певній стадії ЖЦТ обсяг збуту (відповідно, частка на ринку) і обсяг прибутку різний.

Переваги і недоліки різних інструментів роботи з товарним асортиментом наведені в таблиці 1.1.

Таблиця 1.1 – Переваги та недоліки інструментів роботи з товарним асортиментом

Інструмент	Переваги	Недоліки
Правило Парето	Легкий у використанні. Універсальний.	Пропорція 80/20 не є конкретною величиною і може відхилятися до $\pm 20\%$ .
АВС-аналіз	Дозволяє визначити найбільш пріоритетні позиції в асортименті і виділити аутсайдерів.	Не дозволяє оцінювати сезонні коливання продажів. Не може застосовуватися в магазинах з щомісячним оновленням асортименту (наприклад, бутіки модного одягу), так як передбачає наявність статистики продажів за кілька (не менше 3) місяців.

Продовження таблиці 1.1

XYZ-аналіз	Дозволяє прогнозувати стабільність продажів і рівень споживання.	Зовнішні фактори (сезонність попиту, регулярність поставок, коливання цін та ін.) можуть викликати значні коливання продажів і, як наслідок, високі показники коефіцієнта варіації.  Складний в застосуванні для сезонних товарів.
Поєднаний ABC- і XYZ-аналіз	Дозволяє підвищити ефективність управління товарними запасами. Дозволяє виробити оптимальну закупівельну політику.	Не вживають для сезонних товарів і товарів, що мають короткий ЖЦ. Обов'язкова наявність чіткого обліку всіх товарів і статистики продажів.
Матриця Ансоффа	Дає наочне уявлення базисних стратегій.	Орієнтація тільки на зростання. Обмеження: дві змінні (продукт і ринок), кожна з яких може приймати тільки два значення - діючий і новий, що дуже проблематично, якщо інші умови мають значення для успіху.
Матриця BCG	Можливість відстежувати траєкторії руху товарів в матриці і коригувати асортиментну політику.	Відсутність критеріїв, які відносять товари до тієї чи іншої групи. Складність в отриманні даних для побудови матриці.
Матриця GE	Може застосовуватися в будь-яких ринкових умовах.	Невелика кількість стратегій поведінки фірми на ринку (розширення, захист або збір врожаю) є статичною моделлю.



Концепція орієнтації на ЖЦТ	Можливість передбачити поведінку товару на ринку в залежності від стадії ЖЦ.	Не всі товари проходять 4 стадії ЖЦ (наприклад, сезонні товари). Відсутність критеріїв, що визначають перехід з однієї стадії в іншу.  Відсутність орієнтації на прибутковість товарів.
-----------------------------------	---	---

Всі прийоми засновані, в цілому, на дослідженні ринкового сприйняття продукції і на оцінці продукції з точки зору економічної ефективності підприємства.

Практично всі інструменти припускають наявність тривалої статистики про продажі по кожній групі товарів, а також чіткого обліку товарної номенклатури, що не завжди можливо на роздрібних підприємствах [14].

### 1.3 Актуальність проблеми і постановка задачі

Побудова стратегій розвитку вимагає глибокого аналізу даних та виявлення прихованих закономірностей у діяльності підприємств роздрібної торгівлі. Сьогодні підприємства збирають і накопичують детальну інформацію про кожну окрему покупку, використовуючи кредитні картки магазину і комп'ютеризовані системи контролю і т.п. При цьому зі зниженням питомої вартості зберігання даних, зростає обсяг інформації, в результаті чого виникають завдання, пов'язані з необхідністю обробки великих масивів даних з метою пошуку нових закономірностей, встановлення і виявлення нових знань.

Для аналізу та обробки даних широко застосовуються методи і засоби штучного інтелекту, зокрема нейронні мережі, нечіткі моделі, дерева рішень, байєсові мережі, методи регресійного аналізу та ін. Однак такі методи, як

правило, використовуються для обробки структурованих даних, представлених у вигляді масивів, що містять значення ознак і вихідних параметрів примірників вибірки.

В даний час спостерігається надлишок так званих неструктурованих даних, в яких кожна одиниця зберігання не може бути представлена скінченним числом ознак (інформація про товари, транзакції, покупця і т.п.). Такі дані подаються, як правило, у вигляді послідовностей пов'язаних подій. Крім того, розмір кожної транзакції (множини подій, що сталися одночасно) не є фіксованим.

У зв'язку з цим виникають завдання:

- скорочення обсягів неструктурованих даних шляхом видалення надлишкових транзакцій, виключення яких з подальшого розгляду не вплине на якість синтезованих правил і моделей;
- виявлення невідомих правил, що дозволяють отримувати нові знання на основі наявних неструктурованих даних;
- побудови моделей на основі великих масивів неструктурованих даних для вирішення практичних завдань прогнозування, класифікації та кластеризації даних.

Завдяки зібраній інформації, система підтримки прийняття рішень на основі інтелектуального аналізу даних допоможе роздрібним підприємствам у вирішенні наступних задач:

1. Націлювання клієнтів. Індивідуальна поведінка клієнтів може бути визначена на основі великих масивів даних. Аналізуючи кожного клієнта окремо, можна персоналізувати рекомендації щодо продуктів, для підвищення задоволення споживача.

2. Керування запасами. Методи інтелектуального аналізу даних можуть покращити керування запасами на складі та в магазині. Наприклад, поєднання даних історії збуту та сезонних продажів може покращити прогнозування потреби в поповненні запасів та передбачати зміну попиту споживачів. Також,

роздрібні торговці можуть автоматизувати поповнення запасів, що дозволить зменшити затримки в їх постачанні.

3. Оптимізація цін. Різноманітні джерела даних можуть бути використаними в прийнятті рішень щодо ціноутворення. Деталізація даних продажу може бути використана для аналізу попиту ринку при зміні цін або зміні продуктів, що потім може бути використано для отримання оптимального рішення щодо визначення ціни товару.

4. Аналіз споживчого кошика призначений для виявлення товарів, які покупці прагнуть купувати разом. Знання споживчої кошики необхідне для покращення реклами, побудови стратегії створення запасів товарів і способів їх розміщення в торгових залах.

5. Дослідження часових шаблонів дозволяє торговим підприємствам приймати рішення про те, через який час доцільно пропонувати клієнту купити наступний товар, що має відношення до товару, який він купив сьогодні.

6. Створення моделей прогнозування дає можливість торговим підприємствам дізнаватись про характер потреб різних категорій клієнтів із визначеною поведінкою. Ці знання необхідні для розробки точно націлених економічних заходів для продажу окремих категорій товарів [15].

### Висновки до розділу

Отже, керівництво роздрібних підприємств має необхідність у виборі оптимальних рішень під час здійснення маркетингової діяльності. Для забезпечення обґрунтованості, вони мають прийматися на основі аналізу даних, що були зібрані впродовж усієї діяльності підприємства. В свою чергу аналіз даних роздрібною торгівлі поділяється на описову, приписну та

інтелектуальну аналітики. Створення системи підтримки прийняття рішень, що буде базуватися на комплексному використанні аналітик, дозволить приймати обґрунтовані рішення в процесі управління підприємством.

## РОЗДІЛ 2 ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ У РОЗДРІБНІЙ ТОРГІВЛІ

### 2.1 Побудова асоціативних правил

Для обробки великих масивів неструктурованих даних і вирішення зазначених завдань доцільно використовувати методи пошуку асоціативних правил, що дозволяють виявляти нові закономірності виду "якщо умова, то дія" в наявних даних і синтезувати на їх основі інтерпретовані бази правил, зрозумілі експертам в прикладних областях [16].

В даний час запропоновано досить велика кількість видів асоціативних правил, кожен з яких доцільно застосовувати для вирішення певного класу задач. Тому, проведемо огляд і класифікацію асоціативних правил для подальшого їх застосування з метою вирішення практичних задач інтелектуального аналізу даних у роздрібній торгівлі.

#### 2.1.1 Синтез асоціативних правил

Нехай задано набір даних  $D$ :

$$D = \{T_1, T_2, \dots, T_{N_D}\},$$

що представляє собою базу даних транзакцій, у якій кожен елемент  $T_j, j = 1, 2, \dots, N_D$  містить інформацію про деякі взаємозв'язані події, де  $N_D = |D|$  – кількість елементів (транзакцій) в наборі даних  $D$ .

Елементи  $T_j$  можуть представлятись у вигляді:

$$T_j = (tid_j, item_j),$$

де  $tid_j$  – ідентифікатор  $j$ -ї транзакції  $T_j$ ;  $item_j = \{t_{1j}, t_{2j}, \dots, t_{N_{item_jj}}\} \subseteq I$  – список елементів транзакції  $T_j$ ;  $t_{ij}$  –  $i$ -й елемент списку  $item_j$ ,  $i = 1, 2, \dots, N_{item_j}$ ;  $N_{item_j} = |item_j|$  – кількість елементів множини  $item_j$ ;  $I = \{\tau_1, \tau_2, \dots, \tau_{N_I}\}$  – множина можливих значень, які можуть входити в список елементів  $item_j$  кожної транзакції  $T_j$ ,  $j = 1, 2, \dots, N_T$  набору даних  $D$ ;  $\tau_a$  –  $a$ -й елемент множини  $I$ ,  $a = 1, 2, \dots, N_I$ ;  $N_I = |I|$  – кількість елементів в  $I$ .

Таким чином, кожна транзакція  $T_j$  набору даних  $D$  є списком елементів  $item_j$ , що є підмножиною множини  $I$ .

Асоціативним правилом (АП) називається імплікація  $X \rightarrow Y$ , в якій набори  $X$  та  $Y$  не перетинаються:

$$X \rightarrow Y: X \subset I, Y \subset I, X \cap Y = \emptyset$$

Тобто асоціативне правило описує залежності виду: «Із події  $X$  випливає подія  $Y$ » або «якщо умова, то дія».

Задача пошуку асоціативних правил полягає в тому, щоб на основі набору даних  $D$  (бази даних транзакцій) знайти закономірності між подіями  $\tau_a \in I$ ,  $a = 1, 2, \dots, N_I$ .

Задача побудови АП зв'язана із необхідністю обчислення підтримки достовірності правил  $a$ . Набір  $X \subset I$  із бази  $D$  має підтримку  $supp(X)$ , що визначається як відношення кількості транзакцій  $T$  в наборі даних  $D$ , що містять множину елементів  $X$ , до загальної кількості транзакцій в базі даних  $D$ .

Підтримкою  $supp(X \rightarrow Y)$  правила  $X \rightarrow Y$  є підтримка множини  $X \cup Y$ :  $supp(X \rightarrow Y) = supp(X \cup Y)$ .

Достовірністю  $conf(X \rightarrow Y)$  правила  $X \rightarrow Y$  називають відношення його підтримки  $supp(X \rightarrow Y)$  до підтримки  $supp(X)$  множини  $X$ .

Процес синтезу асоціативних правил може бути розділеним на два етапи:

1. Генерування всіх наборів  $X$  з рівнем підтримки не нижче заданого експертом порогового значення  $minsupp(X)$  в результаті чого формуються найчастіші набори  $X \subset I$ .

2. Генерування всіх правил  $X \rightarrow Y$  із рівнем достовірності не нижче заданого експертом порогового значення  $minconfidence(X \rightarrow Y)$  [17].

### 2.1.2 Аналіз видів асоціативних правил

При пошуку взаємозв'язків між різними елементами в базах даних транзакцій  $D = \{T_1, T_2, \dots, T_{N_T}\}$  часто необхідно знаходити не лише позитивні асоціативні правила  $X \rightarrow Y$ , а й інші види правил. До таких правил відносяться: негативні, числові, узагальнені, часові та нечіткі асоціативні правила.

Негативні АП характеризують від'ємний взаємозв'язок між різними подіями типу: «Якщо відбулася подія  $X$ , то подія  $Y$  не наступить» ( $X \rightarrow \bar{Y}$ ) або «Якщо не відбулася подія  $X$ , то подія  $Y$  відбудеться» ( $\bar{X} \rightarrow Y$ ).

Необхідність отримання негативних асоціативних правил ( $X \rightarrow \bar{Y}$ ) та ( $\bar{X} \rightarrow Y$ ) разом з позитивними правилами  $X \rightarrow Y$  пояснюється наступним чином. Побудова повного набору асоціативних правил ( $X \rightarrow Y, X \rightarrow \bar{Y}, \bar{X} \rightarrow Y$ ) між різноманітними об'єктами  $\tau_a \in I, a = 1, 2, \dots, N_I$  бази даних  $D = \{T_1, T_2, \dots, T_{N_T}\}$  дозволить більш детально описати досліджувані залежності,





синтезу АП, що дозволяють добувати як позитивні так і негативні асоціативні правила.

Часті ознаки  $\tau_a \in I, a = 1, 2, \dots, N_I$  можуть приймати не лише бінарні, а й числові значення із деякого діапазону значень  $\tau_a \in [\tau_{amin}; \tau_{amax}]$  або множин значень  $T(\tau_a) = \{\tau_{a1}, \tau_{a2}, \dots, \tau_{aN_{\tau a}}\}$ .

Тому актуальною задачею є добування правил вигляду «Якщо  $X \in [X_{min}; X_{max}]$ , то  $Y \in [Y_{min}; Y_{max}]$ »

Числовим асоціативним правилом називається імплікація вигляду:

$$\langle X, v(X) \rangle \rightarrow \langle Y, v(Y) \rangle$$

де  $v(X) \in T(X)$  і  $v(Y) \in T(Y)$  - значення змінних  $X$  і  $Y$ , відповідно, що належать множинам можливих значень  $T(X)$  і  $T(Y)$ .

Підтримка  $supp(X \rightarrow Y)$  числового АП  $X \rightarrow Y$  визначається за формулою:

$$supp(X \rightarrow Y) = \frac{1}{|D|} \sum_{i=1}^{|D|} \prod_{j=1}^{N_X+N_Y} v_j(\tau_a), \tau_a \in (X \cup Y),$$

де  $N_X$  і  $N_Y$  - кількість елементів  $\tau_a \in I$  в множинах  $X = \{\tau_1, \tau_2, \dots, \tau_{N_X}\}$  і  $Y = \{\tau_{N_X+1}, \tau_{N_X+2}, \dots, \tau_{N_X+N_Y}\}$  відповідно;  $v_j(\tau_a)$  - значення  $a$ -ї ознаки  $\tau_a$  в  $j$ -й транзакції  $T_j$  бази даних  $D$ .

Достовірність  $conf(X \rightarrow Y)$  числового АП  $X \rightarrow Y$  визначається аналогічно позитивним АП. При цьому підтримка  $supp(X)$  множини  $X$  обчислюються у відповідності з формулою:

$$supp(X) = \frac{1}{|D|} \sum_{i=1}^{|D|} \prod_{j=1}^{N_X} v_j(\tau_a), \tau_a \in X.$$

Процес пошуку числових АП по заданим наборам даних  $D$  пов'язаний з необхідністю розбиття на інтервали діапазонів можливих значень елементів  $\tau_a \in I$ , що входять в транзакції  $T_j, j = 1, 2, \dots, N_T$ . В результаті такого розбиття кожна  $j$ -а транзакція  $T_j = (tid_j, item_j)$  представляється списком елементів  $item_j = \{t_{1j}, t_{2j}, \dots, t_{N_{item_jj}}\} \subseteq I$ , в якому кожний  $i$ -й елемент  $t_{ij}$  представляється у вигляді

$$t_{ij} = (\text{елемент } \tau_a \in I; \text{діапазон значень елемента } \tau_a)$$

При цьому множина  $I = \{\tau_1, \tau_2, \dots, \tau_{N_I}\}$  можливих значень, які можуть входити в список елементів  $item_j$  кожної транзакції  $T_j$ , містить елементи  $\tau_a$ :

$$\tau_a \in [\tau_{a \min}; \tau_{a \max}] = \bigcup_{c=1}^{N_{\text{розб}\tau_a}} [\tau_{a \min c}; \tau_{a \max c}], a = 1, 2, \dots, N_I,$$

де  $\tau_{a \min}$  і  $\tau_{a \max}$  - мінімальне і максимальне значення, яке може приймати  $a$ -й елемент  $\tau_a$  множини  $I$ ;  $\tau_{a \min \tilde{n}}$  і  $\tau_{a \max \tilde{n}}$  - мінімальне і максимальне значення  $c$ -го інтервалу розбиття значень  $a$ -го елемента  $\tau_a$  множини  $I$ ;  $N_{\text{розб}\tau_a}$  - кількість інтервалів розбиття  $a$ -го елемента  $\tau_a$ .

Після дискретизації значень числових змінних виконується пошук асоціативних правил  $X \rightarrow Y$ . При цьому використовуються методи добування асоціативних правил, що задовольняють наведеним вище умовам до позитивних правил виду  $X \rightarrow Y$ , але при такому пошуку кожен діапазон дискретизації кожної змінної вважається окремим елементом, котрий може бути використаний при побудові асоціативного правила.

Однак необхідність дискретизації діапазонів значень змінних для добування числових АП суттєво збільшує простір пошуку і вимоги до обчислювальних ресурсів ЕОМ. Крім того, в деяких випадках дискретизація призводить до невдалих розбиттів діапазону значень змінних, в результаті

чого не забезпечується прийнятною точністю прогнозування або класифікації по синтезованій базі АП. Тому актуальним завданням є розробка нових методів пошуку чисельних АП, вільних від зазначених недоліків.

Іноді елементи  $\tau_a \in I$ , що утворюють транзакції  $T_j$  можуть бути об'єднані у групи, а групи елементів можуть формувати групи більш високого рівня, формуючи тим самим деяку ієрархічну деревоподібну структуру. В таких випадках варто добувати правила, що зв'язують не лише набори конкретних елементів  $\tau_a \in I$ , але й елементи з групами, а також групи з групами. Це особливо корисно в задачах аналізу взаємозв'язку споживчого попиту на різні групи товарів.

Нехай задана база даних транзакцій  $D$ , що містить елементи  $\tau_a \in I$ , які можуть бути віднесені до визначених груп  $Gr = \{gr_1, gr_2, \dots, gr_{N_{Gr}}\}$ . Тобто може бути побудоване дерево, що описує зв'язки типу «частина - ціле» досліджуваної предметної області. Для вузла нащадку  $\tau_{child}$  будемо рахувати вузол  $\tau_{parent}$ , котрий знаходиться на більш високому рівні дерева від якого існує шлях до вузла  $\tau_{child}$ .

Узагальненим асоціативним правилом називається імплікація  $X \rightarrow Y$ , в котрій ні один із елементів множини  $Y = \{\tau_{Y1}, \tau_{Y2}, \dots, \tau_{N_Y}\}$  не є предком будь-якого елемента з множини  $X = \{\tau_{X1}, \tau_{X2}, \dots, \tau_{N_X}\}$ . При цьому інші умови для АП зберігаються.

Задача пошуку узагальнених АП полягає у тому, щоб на основі заданої бази даних і побудованої ієрархії елементів ідентифікувати всі правила з рівнями підтримки, достовірності і цікавості, не нижче заданих порогових значень  $minsupport$ ,  $minconfidence$  і  $\epsilon_I$ . При цьому підтримка  $supp(X \rightarrow Y)$  і достовірність  $conf(X \rightarrow Y)$  узагальнених АП  $X \rightarrow Y$  обчислюється аналогічно позитивним АП.

Використання додаткової інформації про ієрархічні зв'язки і про можливості групування елементів  $\tau_a \in I$ , а також введення додаткової

множини  $Gr = \{gr_1, gr_2, \dots, gr_{N_{Gr}}\}$  при пошуку АП дозволяє добувати правила між різними ієрархічними рівнями, що в свою чергу, забезпечує можливість виявлення прихованих зв'язків між різними наборами елементів. Це досягається в тому числі і за рахунок того, що підтримка групи  $supp(gr)$  може бути більшою підтримки елементів  $\tau_a \in I$ , що її формують, і відповідно бути більшою мінімальної підтримки  $minsupport(X)$ , що дозволить синтезувати правила типу  $gr \rightarrow Y, X \rightarrow gr, gr_1 \rightarrow gr_2$ , в той час, як деякі правила типу  $X \rightarrow Y$  не будуть добути в силу не виконання умов відповідності, умов задоволення пороговим значенням підтримки і достовірності [19].

З ціллю добування узагальнених асоціативних правил застосовують методи пошуку позитивних АП. При цьому кожна транзакція  $T_j$  бази даних  $D$  розширюється шляхом доповнення її всіма предками кожного з елементів, що входить у неї. Але застосування даного підходу викликає наступні проблеми:

- елементи  $gr$  що розміщені на верхніх ієрархічних рівнях дерева, характеризуються суттєво більш високими значеннями підтримки, що, як правило, призводить до їхньої появи в більшості синтезованих правил, і, відповідно, до побудови надлишкових правил;

- суттєве збільшення простору пошуку, зв'язане із додаванням групових елементів  $gr$  в транзакції  $T_j$ , що ускладнює задачу добування АП і збільшує кількість добутих правил.

Необхідність усунення приведених недоліків зумовлює потребу створення розробки нових і модифікації існуючих методів добування асоціативних правил.

При роботі з базами даних, що містять інформацію про події, пов'язані з часом, доцільно синтезувати часові асоціативні правила. Суттєвою відмінністю під попередньо згаданих баз даних є наявність інформації про час транзакції  $T_j$ . Тому більшість понять і визначень, наведених для позитивних АП є вірними і для часових АП. Виключенням є підтримка правил.

До часових АП відносяться, АП, що описують залежності, зв'язані з деякими інтервалами часу. Такі правила можуть бути представлені у вигляді: «В інтервал часу  $time(X)$  істинним є вираз: Якщо  $X$ , то  $Y$ ». Часовим інтервалом  $time(X)$  набору елементів  $X$  вважається інтервал часу, протягом якого виконується цей набір:

$$time(X) = [time(T_f); time(T_e)],$$

де  $time(T_f)$  і  $time(T_e)$  - час першої  $T_f$  і останньої  $T_e$  транзакцій, що містять набір елементів  $X$ ,  $time(T_f) < time(T_e)$ .

При такому підході підтримкою  $supp(X)$  набору елементів  $X$  є відношення кількості  $N_{T_j \in D | X \subseteq T}$  транзакцій  $T_j$ , що містять набір  $X$ , до кількості  $N_{T_j \in D | time(T) \subseteq time(X)}$  транзакцій, що відбуваються в часовому інтервалі  $time(X)$ . Підтримка  $supp(X)$  в такому випадку обчислюється за формулою:

$$supp(X) = \frac{N_{T_j \in D | X \subseteq T}}{N_{T_j \in D | time(T) \subseteq time(X)}}$$

Також до часових АП відносяться циклічні АП, що описують регулярні циклічні дії в часі. Наприклад, до таких АП можуть бути віднесені правила, які є істинними в визначений проміжок часу кожного дня. Такі правила можуть бути представлені у вигляді: «В інтервал часу  $time(X)$  з періодичністю  $Period$  істинним є вираз: Якщо  $X$ , то  $Y$ ».

Нехай  $\Delta t$  - деякий інтервал часу. Тоді транзакція  $T_j$  виконується в  $j$ -й інтервал часу  $time(T_j): [j \cdot \Delta t; (j + 1) \cdot \Delta t]$ . Позначимо  $ST(time(T_j))$  - множину транзакцій, що виконуються в  $j$ -й інтервал часу  $time(T_j): ST(time(T_j)) = \{T \in D | time(T) \subseteq time(T_j)\}$ . Циклічна підтримка

$supp(X \rightarrow Y, time(T_j))$  АП  $X \rightarrow Y$  в  $j$ -й інтервал часу  $time(T_j)$  відношення кількості транзакцій в множині  $ST(time(T_j))$  до загальної кількості транзакцій в базі даних. Аналогічним чином визначається достовірність  $conf(X \rightarrow Y, time(T_j))$  правила  $X \rightarrow Y$  в  $j$ -й інтервал часу:

$$conf(X \rightarrow Y, time(T_j)) = \frac{supp(X \rightarrow Y, time(T_j))}{supp(X, time(T_j))}$$

Визначимо цикл як пару  $c = (Period, O)$ , в якій  $Period$  - період або довжина циклу,  $O$  — зміщення, тобто конкретний час спрацьовування правила,  $0 \leq O \leq Period$ . Таким чином, АП  $X \rightarrow Y$  є циклічним із циклом  $c = (Period, O)$ , якщо воно спрацьовує в кожний  $O$ -й інтервал часу з періодичністю  $Period$ .

Для добування часових АП, аналогічно пошуку інших видів правил, задаються порогові значення мінімальної підтримки, достовірності і цікавості правила:  $minsupport$ ,  $minconfidence$  і  $\varepsilon_I$  відповідно.

Однак застосування методів пошуку позитивних АП для добування часових АП пов'язане з проблемою обчислення цікавості правил із урахуванням часу їх виконання. Крім цього, існують проблеми добування часових АП із різними періодами виконання правил, пошуку оптимальних інтервалів дискретизації часу  $\Delta t$ .

Нехай база даних транзакцій характеризується набором транзакцій  $T_j$ , що складаються з елементів  $\tau_a \in I = \{\tau_1, \tau_2, \dots, \tau_{N_I}\}$ , а також невід'ємних вагових коефіцієнтів  $W = \{w_1, w_2, \dots, w_{N_I}\}$ , де  $w_a$  - вага елемента  $\tau_a, a = 1, 2, \dots, N_I$ .

Поняття нечітких АП зв'язано з числовими правилами. При цьому основні поняття і визначення відносно бінарних позитивних АП поширюються на нечіткі АП для числових транзакцій. Числові значення

елементів  $\tau_a$  при пошуку числових АП мають бути розбиті на інтервали, що не пересікаються, кожен з яких розглядається далі, як окремий атрибут або терм. З ціллю усунення проблеми неефективного розбиття на інтервали при добування АП у випадку наявності в базі даних транзакцій числових змінних використовується теорія нечітких множин, в поєднанні з теорією АП, що дозволяє добувати АП і будувати на їх основі нечіткі бази правил.

Нечітка підтримка  $supp(X, T_j)$  набору  $X$  в транзакції  $T_j$  визначається за формулою

$$supp(X, T_j) = \bigcap_{i=1}^{|X|} \mu_{X_i}(T_j),$$

де  $\mu_{X_i}(T_j)$  - функція належності змінної  $X_i$  множині  $T_j$ .

Нечітка підтримка  $supp(X)$  набору  $X$  по всій базі даних транзакцій може бути обчислена аналогічно за формулою:

$$supp(X) = \sum_{T_j: X \subseteq T_j} supp(X, T_j) = \prod_{i=1}^{|X|} \mu_{X_i}(T_j)$$

При добуванні нечітких АП використовуються поняття зваженої нечіткої підтримки набору елементів  $X$  і правила  $X \rightarrow Y$ .

Зважена нечітка підтримка  $wsupp(X)$  набору елементів  $X$  визначається як добуток підтримки  $supp(X)$  на суму вагових коефіцієнтів  $w_a$  елементів  $\tau_a$ , що присутні в наборі  $X$ :

$$wsupp(X) = \left( \sum_{\tau_a \in X} w_a \right) supp(X)$$

Зважена нечітка підтримка правила  $X \rightarrow Y$  обчислюється за формулою

$$wsupp(X \rightarrow Y) = \left( \sum_{\tau_a \in X \cup Y} w_a \right) supp(X \cup Y)$$

Зважена нечітка достовірність правила  $X \rightarrow Y$  обчислюється за формулою

$$wconf(X \rightarrow Y) = \frac{wsupp(X \rightarrow Y)}{wsupp(X)} = \frac{(\sum_{\tau_a \in X \cup Y} w_a) supp(X \cup Y)}{(\sum_{\tau_a \in X} w_a) supp(X)}$$

Для добування нечітких АП задаються рівні мінімальної зваженої підтримки  $wminsupport$ , мінімальної зваженої достовірності  $wminconfidence$  і мінімальної цікавості правила  $\varepsilon_I$ . Якщо зважена нечітка підтримка  $wsupp(X)$  набору елементів  $X$  не менша мінімально допустимого значення ( $wsupp(X) \geq wminsupport$ ), то набір  $X$  вважається частим. Аналогічно, якщо значення зваженої нечіткої достовірності  $wconf(X \rightarrow Y)$  правила  $X \rightarrow Y$  не менше мінімально допустимого значення ( $wconf(X) \geq wminconfidence$ ), то правило  $X \rightarrow Y$  вважається достовірним. Крім того, рівень цікавості правила має бути не меншим заданого порогового значення  $\varepsilon_I$ .

Методи добування нечітких АП на початковому етапі виконують перетворення кожної числової змінної до нечіткої множини з відповідними лінгвістичними термами, використовуючи для цього функції належності. Далі виконується розрахунок скалярної потужності кожного лінгвістичного терму по всій базі даних  $D$ , а також обчислення підтримки наборів елементів, використовуючи ітеративний підхід для пошуку великих наборів даних.



Потім виконується пошук нечітких АП з цих великих наборів даних. Кожен елемент використовується тільки в лінгвістичному термі з максимальною потужністю на більш пізніх ітераціях, в результаті чого кількість оброблюваних нечітких областей така ж, як кількість елементів  $\tau_a$  в множині  $I = \{\tau_1, \tau_2, \dots, \tau_{N_1}\}$ . Таким чином, основний акцент здійснюється на найбільш важливі лінгвістичні терми, що знижує часову складність таких методів.

Однак, необхідність виділення нечітких термів при добуванні АП потребує розв'язання задачі кластерного аналізу (в результаті чого терми можуть бути визначені як проекції границь кластерів на осі ознак), що ускладнює процес пошуку нечітких АП, або участь користувача-експерта в прикладній області, що зменшує рівень автоматизації програмних засобів добування нечітких АП на основі відповідних методів [20].

### 2.1.3 Алгоритм Apriori

Визначення частих наборів об'єктів – операція, що потребує великої кількості обчислень і, відповідно, часу. Алгоритм Apriori використовує одну із властивостей підтримки: підтримка будь-якого набору об'єктів не може перевищувати підтримки будь-якого із його підмножин:

$$Supp_F \leq Supp_E, E \subset F.$$

Алгоритм Apriori визначає набори, що часто зустрічаються за декілька етапів. На  $i$ -му етапі визначаються всі часті  $i$ -елементні набори. Кожен етап складається із двох кроків: формування кандидатів і підрахунку підтримки кандидатів.

Розглянемо  $i$ -й етап. На кроці формування кандидатів алгоритм створює множину кандидатів із  $i$ -елементних наборів, чия підтримка поки, що не обраховується. На кроці підрахунку кандидатів алгоритм сканує множину транзакцій, обчислюючи підтримку наборів-кандидатів. Після сканування відкидаються кандидати, підтримка яких менша визначеного користувачем мінімуму і зберігаються лише часті  $i$ -елементні набори.

Опишемо позначення, що використовуються в алгоритмі:

$L_k$  — множина  $k$ -елементних частих наборів, чия підтримка не менша заданої користувачем. Кожен член множити має набір впорядкованих ( $i_j < i_p$ , якщо  $j < p$ ) елементів  $F$  і значення підтримки набору  $Supp_F > Supp_{min}$ :

$$L_k = \{(F_1, Supp_1), (F_2, Supp_2), \dots, (F_q, Supp_q)\},$$

де  $F_j = \{i_1, i_2, \dots, i_k\}$ ;

$C_k$  — множина кандидатів  $k$  — елементних наборів потенційно частих. Кожен член множини має набір впорядкованих ( $i_j < i_p$ , якщо  $j < p$ ) елементів  $F$  і значення підтримки набору  $Supp$ .

Опишемо даний алгоритм покроково:

Крок 1. Присвоїти  $k = 1$  і виконати відбір усіх 1-елементних наборів, у котрих підтримка більша мінімально заданого користувачем значення  $Supp_{min}$ .

Крок 2.  $k = k + 1$ .

Крок 3. Якщо не вдається створити  $k$ -елементні набори, то завершити алгоритм, інакше перейти на наступний крок.

Крок 4. Створити множину  $k$ -елементних наборів кандидатів в часті набори. Для цього необхідно об'єднати в  $k$ -елементні кандидати  $(k - 1)$  — елементні часті набори. Кожен кандидат  $c \in C_k$  буде формуватися шляхом

додавання до  $(k - 1)$  – елементному частому набору –  $p$  елемента із іншого  $(k - 1)$  – елементного частого набору –  $q$ . При чому додається останній елемент набору  $q$ , який по порядку вище, чим останній елемент набору  $p$ . При цьому усі перші  $k - 2$  елементи обох наборів однакові.

Крок 5. Для кожної транзакції  $T$  із множини  $D$  вибрати кандидатів  $C_t$  із множини  $C_k$ , що присутні в транзакції  $T$ . Для кожного набору із побудованої множини  $C_k$  видалити набір, якщо хоча б одна із його  $(k - 1)$  підмножин не є частим, тобто відсутній в множині  $L_{k-1}$ .

Крок 6. Для кожного кандидата із множини  $C_k$  збільшити значення підтримки на одиницю.

Крок 7. Вибрати лише кандидатів  $L_k$  із множини  $C_k$ , у яких значення підтримки більше заданого користувачем  $Supp_{min}$ .

Результатом роботи алгоритму є об'єднання всіх множин  $L_k$  для всіх  $k$  [21].

## 2.2 Прогнозування у роздрібній торгівлі

Велика кількість проведених досліджень по керуванню товарними запасами, реалізації маркетингової політики, стратегічному плануванню в сфері роздрібної торгівлі присвячено адекватному прогнозування попиту на товар. Значимість прогнозування є незаперечною, тим більше в період бурхливого зростання даних про стан економіки та інших сфер життя сучасного суспільства. Необхідно комплексно підходити до аналізу існуючої на підприємстві системи керування запасами, до оцінки стану інформаційних систем, наявності необхідних даних, а також до виявлення складу факторів, що впливають на попит того чи іншого товару.

При прогнозуванні попиту на конкретний товар, особа, яка приймає рішення, не може знати майбутніх показників для ряду чинників, які є ключовими. Тому від точності моделі прогнозування ключових факторів залежить точність всієї системи прогнозування в цілому. Даний підхід є актуальним для дослідження особливостей функціонування системи руху товарів в роздрібній торгівлі. Моделі прогнозування можуть бути побудовані для двох основних факторів, які впливають на купівельний попит: ендогенний ключовий показник (кількість транзакцій в магазині) і екзогенний (температурний режим). Використання цих показників обумовлене високим ступенем значущості їх впливу на роботу торгової мережі. Кількість транзакцій - це формальний показник ефективності магазину, температурний режим - показник, який найчастіше впливає на розмір купівельного потоку і попит на конкретні види товарів. Методи інтелектуального аналізу даних можуть використовуватися, як в загальному комплексі системи прогнозування товарного попиту на підприємстві роздрібної торгівлі, так і незалежно - для складання необхідних планів і для вироблення необхідних тактичних заходів. Це зумовлено тим, що розглянуті показники несуть важливу інформацію саму по собі: коректне передбачення температурного режиму дозволяє регулювати ціноутворення по деяким видам товарів, а кількість і динаміка транзакцій повідомляє про результати проведених маркетингових заходів [22].

### 2.2.1 Моделі регресійного аналізу

#### 1. Авторегресійна модель.

Авторегресійну модель порядку  $p$  можна записати у наступному вигляді:

$$y(k) = a_0 + a_1 y(k-1) + a_2 y(k-2) + \dots + a_p y(k-p) + \varepsilon(k),$$

де  $y(k)$  – рівень часового ряду в момент часу  $k$  (залежна змінна);  $y(k-1), y(k-2), \dots, y(k-p)$  – рівні часового ряду в моменти часу  $k-1, k-2, \dots, k-p$ , відповідно (незалежні змінні);  $a_0, a_1, a_2, \dots, a_p$  – оцінювані коефіцієнти;  $\varepsilon(k)$  – випадкове збурення, що описує вплив змінних не врахованих в моделі.

Коефіцієнт  $a_0$  визначає постійний рівень ряду і зв'язаний з математичним очікуванням  $E$  співвідношенням

$$a_0 = E(1 - a_1 - a_2 - \dots - a_p).$$

## 2. Модель з ковзним середнім.

Модель зі ковзним середнім порядку  $q$  задається рівнянням

$$y(k) = \varepsilon(k) - b_1 \varepsilon(k-1) - b_2 \varepsilon(k-2) - \dots - b_q \varepsilon(k-q),$$

де  $y(k)$  – рівень часового ряду в момент часу  $k$ ;  $\varepsilon(k-i)$  – значення лишків  $i$  часових періодів назад;  $b_1, b_2, \dots, b_q$  – оцінювані коефіцієнти.

Моделі ковзного середнього  $MA$  дають прогноз значень функції  $y(k)$  на основі лінійної комбінації обмеженого числа  $q$  лишків, в той час, як авторегресійні моделі  $AR$  дають прогноз значення  $y(k)$  на основі лінійної функції апроксимації обмеженого числа  $p$  минулих значень  $y(k)$ .

Використання поняття ковзного середнього в даному випадку означає, що відхилення залежної змінної від свого середнього, тобто величина  $y(k) - E$ , є лінійною комбінацією поточних і минулих значень вектору випадкових збурень.

### 3. Модель авторегресії із ковзним середнім.

Авторегресійну модель і модель з ковзним середнім можна скомбінувати. При описі даної комбінації використовується позначення  $ARMA(p, q)$ , де  $p$  – порядок авторегресійної частини моделі,  $q$  – порядок частини ковзного середнього. Модель  $ARMA(p, q)$  має загальний вигляд

$$y(k) = a_0 + a_1y(k-1) + a_2y(k-2) + \dots + a_py(k-p) + \varepsilon(k) - b_1\varepsilon(k-1) - b_2\varepsilon(k-2) - \dots - b_q\varepsilon(k-q).$$

Дана модель дозволяє будувати прогноз, що залежить як від поточного і попередніх значень залежної змінної, так і від поточних і минулих значень величини випадкового збурення [23].

### 4. Модель у вигляді поліноміального тренду.

Модель у вигляді поліноміального тренду можна записати з використанням детермінованих функцій від часу у вигляді:

$$y(k) = a_0 + d_1k + d_2k^2 + \dots + d_mk^m + \varepsilon(k),$$

де  $k$  – дискретний час;  $d_i$  – коефіцієнти рівняння.

### 5. Модель авторегресії з інтегрованим ковзним середнім.

$$\tilde{y}(k) = d^m y(k),$$

де  $dy(k) = y(k) - y(k-1)$ .

$$\tilde{y}(k) = a_0 + a_1\tilde{y}(k-1) + a_2\tilde{y}(k-2) + \dots + a_p\tilde{y}(k-p) + \varepsilon(k) - b_1\tilde{\varepsilon}(k-1) - b_2\tilde{\varepsilon}(k-2) - \dots - b_q\tilde{\varepsilon}(k-q)$$

### 2.2.2 Критерії адекватності моделі

Прогнози стають важливими компонентами процесу прийняття рішень.

Фірма швидко відмовиться від використання прогнозів, незалежно від складності використовуваних при їх створенні методів, якщо ці прогнози не матимуть необхідної точності. Про точність прогнозу прийнято судити за величиною похибки (помилки) прогнозу - різниці між прогнозованим і фактичним значенням (реалізацією) досліджуваної змінної.

Для забезпечення точності і достовірності результатів прогнозування необхідна перевірка адекватності або верифікація прогнозної моделі.

Перевірка адекватності моделі виконується з використанням формальних статистичних критеріїв. Однак така перевірка проводиться за наявності надійних статистичних параметрів як оригіналу (об'єкта прогнозування), так і моделі. Якщо з якихось причин такі оцінки відсутні, то здійснюють порівняння окремих властивостей оригіналу і моделі. При цьому спочатку повинна перевірятися істинність реалізованих функцій, потім істинність структури і, нарешті, істинність досягаються при цьому значень параметрів. Для цього крім моделі необхідно мати функціонуючий оригінал, тобто проводити супроводжуюче моделювання.

Розглянемо найбільш популярні критерії адекватності моделі.

1. Коефіцієнт детермінації  $R^2$ . Коефіцієнт  $R^2$  визначається за наступною формулою:

$$R^2 = \frac{var(\hat{y})}{var(y)}$$

де  $var(\hat{y})$  – дисперсія частини часового ряду основної змінної, що описується отриманим рівнянням,  $var(y)$  – вибіркова дисперсія основної

змінної. Для адекватної моделі даний коефіцієнт повинен прямувати до одиниці, тобто:  $R^2 \rightarrow 1$  [24].

## 2. Сума квадратів похибок моделі

$$SSE = \sum_{k=1}^N [\hat{y}(k) - y(k)]^2,$$

де  $\hat{y}(k) = \hat{a}_0 + \hat{a}_1 \hat{y}(k-1) + \hat{a}_2 \hat{y}(k-2) + \hat{b}_1 \hat{x}(k) + b_2 z(k)$ ;  $y(k)$  – вимірювання;  $N$  – довжина вибірки. Даний критерій найбільш часто використовується при виборі оптимальної моделі прогнозування в пакетах прикладних програм з прогнозування. Кожне значення помилки прогнозу підводиться до квадрату, що підкреслює великі помилки прогнозу. Ця особливість дуже важлива, оскільки модель прогнозування, що дає середні по величині помилки, в деяких випадках може бути кращою відносно іншої моделі, що має менші помилки, але інколи з недопустимими по абсолютній величині викидами. З можливих кандидатів варто вибирати саме ту модель, для якої  $SSE$  приймає мінімальне значення.

3. Інформаційний критерій Акайке (AIC). Критерій Акайке враховує суму квадратів похибок, кількість вимірів  $N$  і кількість параметрів моделі, що оцінюються  $p$ :

$$AIC = N \ln \left[ \sum_{k=1}^N e^2(k) \right] + 2p.$$

Для кращої моделі критерій має менше значення, оскільки напряду він залежить від суми квадратів похибок. Але також, крім суми квадратів похибок, цей критерій враховує довжину вибірки і кількість оцінюваних параметрів.



4. Статистика Дарбіна-Уотсона (Durbin-Watson). Статистика Дарбіна-Уотсона обчислюється за формулою:

$$DW = 2 - 2\rho,$$

де  $\rho$  – коефіцієнт кореляції між сусідніми значеннями випадкової змінної  $\varepsilon(k) \approx e(k)$ , тобто  $\rho = \text{cov}[e(k)] = E[e(k)e(k-1)]$ .

Цей параметр дозволяє визначити ступінь корельованості похибок моделі. При повній відсутності кореляції між похибками  $DW = 2$ , це найбільше прийнятне значення даного параметра [25].

5. Статистика Фішера  $F$ . Ця статистика визначає ступінь адекватності моделі загалом. Для адекватної моделі виконується умова:

$$F > F_{\text{крит}},$$

де  $F_{\text{крит}}$  визначається по таблиці аналогічно  $t$  – статистиці.

Значення  $F$  пропорційне  $R^2/(1 - R^2)$ , де  $R^2$  – коефіцієнт детермінації.

Тому, моделі вищого ступеня адекватності відповідає більше значення  $F$ .

6. Коефіцієнт Тейла є дуже важливим індикатором точності моделі та її сумісності:

$$U = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i)^2} + \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i)^2}}$$

За побудовою, його величина знаходиться між 0 і 1. Якщо  $U = 1$ , модель не може бути використана для прогнозу. Прогнозовані, на основі отриманої

моделі, і реальні ряди некорельовані. У іншому випадку, якщо  $U = 0$ , то ряди, що були змодельовані та спрогнозовані співпадають з реальними і модель описує дані ідеально [26].

Цей коефіцієнт може бути розкладений на суму відношення упередженості  $U^M$ , відношення варіацій  $U^S$  і відношення коваріацій  $U^C$ .

Відношення упередженості

$$U^M = \frac{(y_i - \hat{y}_i)^2}{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

застосовується для знаходження наявності постійних відхилень для середніх реальних і прогнозованих рядів. Або, інакше кажучи, чи модель весь час завищує прогноз. Чим менша величина  $U^M$ , тим краща модель. Якщо  $U^M = 0$ , то модель високоякісна і в прогнозованих значеннях відсутня упередженість.

$U^S$  – відношення варіацій, яке визначається так:

$$U^S = \frac{(\sigma - \hat{\sigma})^2}{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

Відношення варіацій застосовується для перевірки того, що модель містить достатньо динамічних властивостей для поглинання варіацій реальних рядів.  $U^C$  – відношення коваріацій, яке визначається так:

$$U^C = \frac{2(1 - \rho)(\sigma - \hat{\sigma})^2}{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

Відношення коваріацій  $U^C$  показує, наскільки корельованими є прогнозовані та реальні ряди. Рівність  $U^C = 0$  є свідченням того, що прогнозовані і реальні ряди ідеально корельовано [27].

Три зазначені статистики пов'язані між собою таким чином:

$$U^C + U^M + U^S = 1.$$

#### Висновки до розділу

Отже, методологія інтелектуального аналізу даних містить достатній інструментарій для вирішення задач, що виникають у роздрібній торгівлі. Аналіз асоціативних правил дозволяє дізнатись, які товари мають зв'язок і як варто їх розміщувати у магазині. В свою чергу аналіз і прогнозування часових рядів дає можливість побудувати моделі продажів і дозволяє спрогнозувати кількість проданих товарів у майбутньому.

## РОЗДІЛ 3 РОЗРОБКА ТА ЗАСТОСУВАННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ПІДПРИЄМСТВА РОЗДРІБНОЇ ТОРГІВЛІ

### 3.1 Системи підтримки прийняття рішень

Система підтримки прийняття рішень - це комп'ютерна система, яка за допомогою збору та аналізу великих масивів даних може впливати на процес прийняття рішень організаційного плану в підприємстві і бізнесі. Системи, що інтерактивно взаємодіють із користувачем, дозволяють отримати важливу інформацію з першоджерел, проаналізувати їх і виявити існуючі моделі та залежності для вирішення необхідних завдань.

Із використанням СППР можна слідкувати за всіма доступними інформаційними активами, обрахувати значення обсягів продажів, спрогнозувати дохід організації у майбутньому при гіпотетичному впровадженні певної технології, а також порівняти всі можливі альтернативні рішення.

Система підтримки прийняття рішень - набір математичних методів і моделей, що об'єднані загальною методикою формування альтернатив управлінських рішень в організаційних системах, прогнозування наслідків реалізації кожної альтернативи і обґрунтування вибору найбільш прийняттого управлінського рішення.

Підтримка прийняття рішень полягає у консультуванні особи, що приймає рішення (ОПР). Вона включає:

- допомогу ОПР при аналізі об'єктивної складової, тобто в розумінні і оцінці ситуації, що склалася при обмеженнях із зовнішнього середовища;
- виявлення вподобань ОПР, тобто виявлення і ранжування пріоритетів, врахування невизначеності в оцінках ОПР і формування його вподобань;
- генерацію можливих рішень, тобто формування списку альтернатив;

- оцінку можливих альтернатив, виходячи з вподобань ОПР, і обмеженням зовнішнього середовища;
- аналіз наслідків прийнятих рішень;
- вибір кращого з точки зору ОПР варіанту.

СППР, зазвичай, це інтерактивна комп'ютерна система, яка допомагає ОПР застосовувати дані для ідентифікації рішення і прийняття рішень. Система має мати можливість працювати з інтерактивними запитами за допомогою нескладної для вивчення мови.

СППР володіє наступними основними характеристиками:

- використовує дані і моделі;
- допомагає в прийнятті рішень для неструктурованих і слабо структурованих завдань;
- допомагає прийняти, але не замінює способу генерації рішень менеджерами;
- покращує ефективність рішень.

СППР, що володіє ідеальними характеристиками:

- працює зі слабо структурованими рішеннями;
- призначена для ОПР різного рівня підготовки;
- може використовуватись для групового та особистого використання;
- підтримує наступні фази процесу прийняття рішення: інтелектуальну, проектування і вибір;
- підтримує різні способи та методи вирішення, що допомагає при вирішенні задач групою ОПР;
- є гнучкою та вміє адаптується до різноманітних змін як підприємства, так і його оточення;
- проста у застосуванні і модифікації;
- підтримує як взаємозалежні, паралельні, так і послідовні рішення;
- підтримує еволюційне використання і легко адаптується до мінливих вимог;

- дозволяє використовувати накопичені знання.

### 3.2 Структура даних роздрібного магазину

Структура даних роздрібного магазину може відрізнятися в залежності від системи планування ресурсами, що використовує організація. На рисунку 3.1 зображена загальна схема ієрархії товарів, що використовується в меблевому бізнесі.

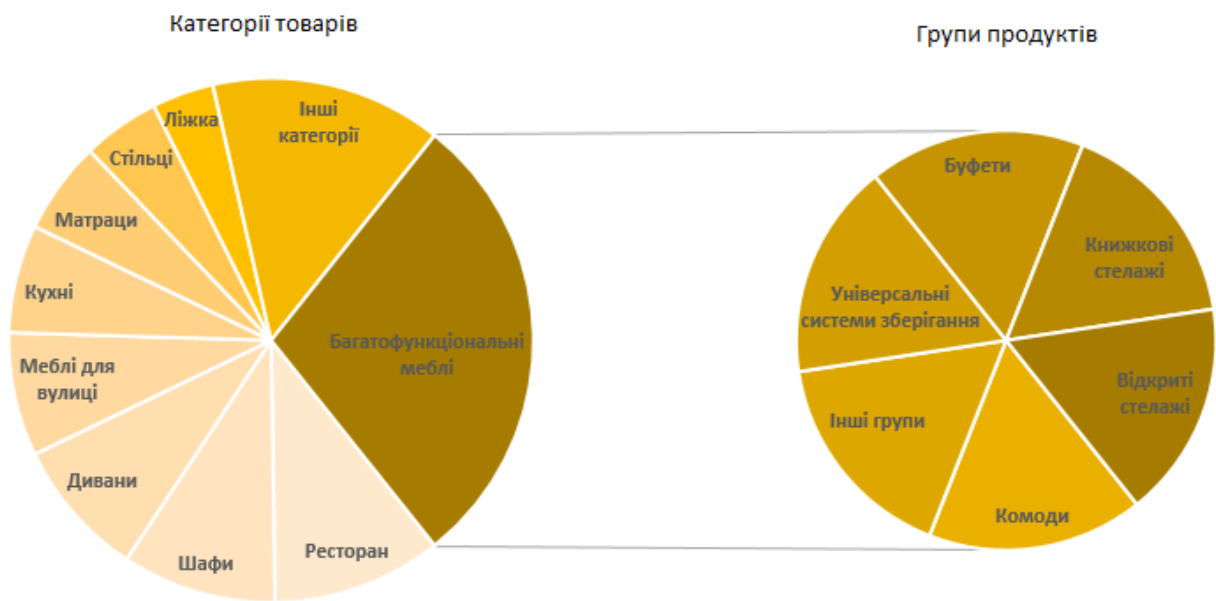


Рисунок 3.1 – Ієрархія товарів в меблевому магазині

Товари у меблевих магазинах входять до груп продуктів, а ті, в свою, чергу до категорій товарів. При введенні нових товарів в систему планування ресурсами, визначається їх приналежність до групи та категорії.

Також зберігається детальна інформація про транзакції, що описана на рис. 3.2.

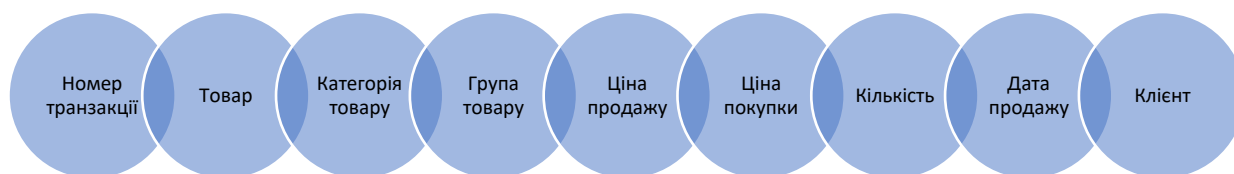


Рисунок 3.2 – Опис транзакції

### 3.3 Структура СППР

Відповідно до блок-схеми СППР, зображеної на рис. 3.3, видно, що архітектура системи передбачає наявність таких основних блоків:

- блок завантаження даних;
- блок попереднього аналізу;
- блок генерації асоціативних правил;
- блок побудови моделей;
- блок прогнозування;
- блок обчислення оцінок;
- блок виведення результатів.

Блок завантаження даних відповідає за імпорт даних, що включають в себе транзакції, категорії товарів, групи продуктів та товари. Для коректного завантаження, файли з даними повинні бути у форматі \*.csv.

Блок попереднього аналізу даних реалізовано на основі ABC-аналізу, що дозволяє визначити категорії, групи та товари, що мають найбільший на прибуток підприємства.

Блок генерації асоціативних правил дозволяє провести асоціативний аналіз категорій та товарів, що дозволяє побачити неочевидні зв'язки між ними.

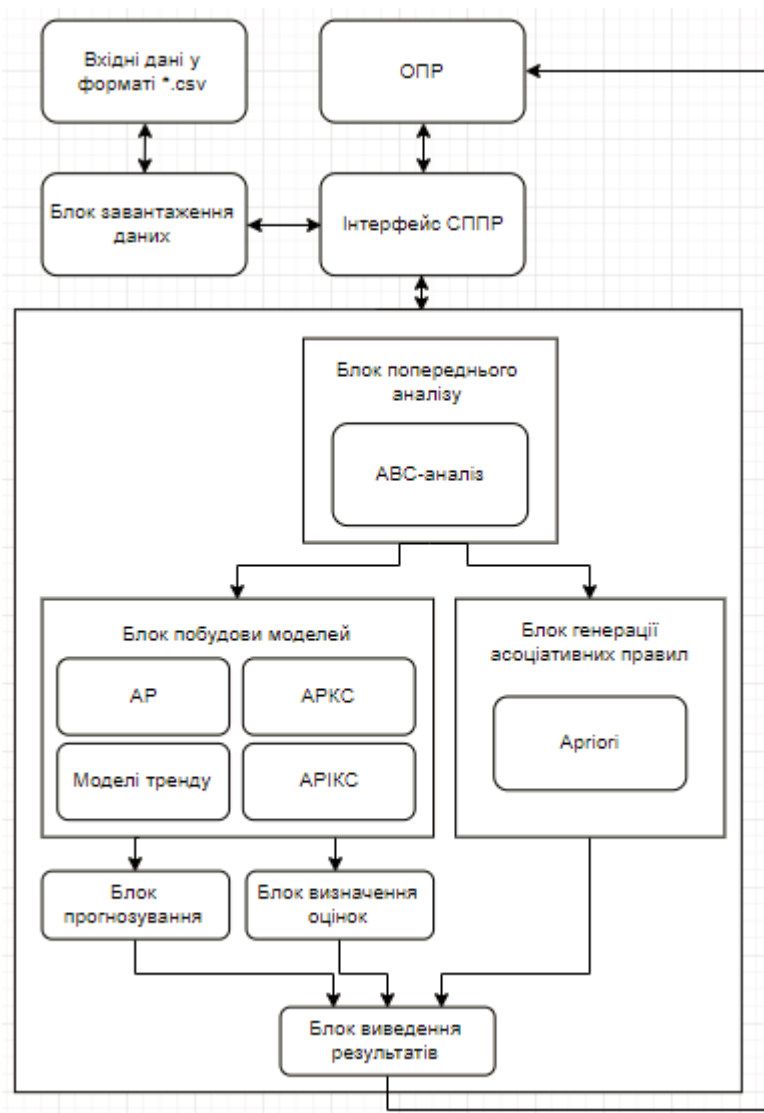


Рисунок 3.3 – Блок-схема СПДР

Блок побудови моделей базується на теорії прогнозування часових рядів та дозволяє будувати наступні моделі: AR, ARMA, ARIMA та модель у вигляді тренду.

Блок прогнозування дозволяє оцінити майбутні продажі та прибуток на основі побудованих моделей.

Блок обчислення оцінок дозволяє побачити, наскільки побудовані моделі є адекватними, а також дозволяє обрати найкращу модель, на основі якої варто робити прогнози.



Блок виведення результатів відповідає за представлення отриманих результатів.

### 3.4 Довідка користувача з експлуатації застосунку

#### 3.4.1 Формат вхідних даних

Вихідні дані поділені на чотири окремих файли та завантажуються у форматі \*.csv. У першому файлі містяться дані з транзакціями за зразком, представленою на рис. 3.4.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Transaction No.	Line No.	Receipt No.	Item No.	Sales Staff	Item Category Code	Product Group Code	Price	Net Price	Quantity	VAT	VAT	Date	Customer No.
1010007738	10000	1,01E+11	11536		76	762	2,29	1,89	-1	CEU	A	6-1-2016	CUST02000320
1010007738	20000	1,01E+11	133038		141	1414	5,99	4,95	-1	CEU	A	6-1-2016	CUST02000320
1010007738	30000	1,01E+11	141401		94	942	0,99	0,82	-1	CEU	A	6-1-2016	CUST02000320
1010007738	40000	1,01E+11	149459		142	1421	4,99	4,12	-1	CEU	A	6-1-2016	CUST02000320
1010007738	50000	1,01E+11	150004		111	1111	4,49	3,71	-1	CEU	A	6-1-2016	CUST02000320
1010007738	60000	1,01E+11	152159		142	1422	3,99	3,3	-1	CEU	A	6-1-2016	CUST02000320
1010007738	70000	1,01E+11	152282		82	822	29,99	24,79	-1	CEU	A	6-1-2016	CUST02000320
1010007738	80000	1,01E+11	165213		93	936	29,99	24,79	-1	CEU	A	6-1-2016	CUST02000320
1010007738	90000	1,01E+11	177225		76	761	2,29	1,89	-1	CEU	A	6-1-2016	CUST02000320
1010007738	100000	1,01E+11	201904		76	762	9,99	8,26	-1	CEU	A	6-1-2016	CUST02000320
1010007738	110000	1,01E+11	204629		73	731	46	38,02	-1	CEU	A	6-1-2016	CUST02000320

Рисунок 3.4 - Приклад таблиці вхідних даних транзакцій

У другому файлі містяться дані з категоріями товарів, зразок файлу зображено на рис. 3.5.

A	B
Code	Description
11	Sofas
12	Armchairs, footstool & sofa ta
13	Armchairs
14	Occasional tables range
15	Solitaire sofa-beds, armchairs
16	Seating & Reclining Furniture
21	Multipurpose furniture
22	Store and display furniture
23	Tv furniture & electronics
24	Living-room Storage
25	Living room tables
26	Secondary Storage
31	Workspaces
32	Work storage furniture.
33	Work seating range
34	Flooring
35	Bedroom furniture
36	Mattresses
37	Bedroom storage
38	Storing Toiletries
41	Beds

Рисунок 3.5 - Приклад таблиці вхідних даних категорій товарів

У третьому файлі містяться дані з групами продуктів, що є логічним розбиттям категорій товарів на групи. Зразок файлу груп продуктів для завантаження зображено на рис. 3.6.

A	B	C
Code	Description	Item Category
111	Sofas	11
112	Leather sofas	11
113	Sofa beds	11
114	Armchairs	11
115	Footstools	11
119	Furniture fabrics	11
121	Sofa tables	12
122	Solitaire sofa beds	12
123	Glass-door cabinets, shelving	12
124	Audio & video furniture - UTG	12
125	Armchairs	12
126	Footstools	12
131	Layer glued armchairs	13
132	Solitaire armchairs	13
133	Natural fibre armchairs	13

Рисунок 3.6 - Приклад таблиці вхідних даних груп продуктів

У четвертому файлі містяться дані про товари. Приклад даних зображено на рис. 3.7.

A	B	H	I	J	K	X	Y
No.	Description	Unit Price	Profit %	Unit Cost	Vendor No.	Item Cate	Product G
137	MEASURING KITCHEN	0	0	0	zero	973	9731
203	KITCHEN PLUMBING	0	0	0	zero	972	9722
496	BATHROOM PLUMBING	0	0	0	zero	972	9722
528	INST IKEA GAS APP	0	0	0	zero	972	9722
533	inst of wrktp	0	0	0	zero	972	9722
552	CONN UPPEVA TV DEV	0	0	0	zero	972	9722
566	HOME DELIV OUTSD REGION	0	0	0	zero	971	9711
571	LOAD SERV VAT SPEC	0	0	0	zero	971	9713
11013	LACK WLL SHLF 190X26 WHITE4	66,1074	86,9152	8,65	VEN02000006	21	215
11428	GRUNDTAL WLL SHLF 80 STAINLESS STEEL	14,04132	61,7771	5,367	VEN02000006	76	762
11536	GRUNDTAL S-HOOK 11 STAINLESS STEEL 5-P	1,89256	63,84791	0,6842	VEN02000006	76	762
15134	SVALKA WHI WNE GLS 25 CL CLEAR GLASS 6-P	2,8843	27,53874	2,09	VEN02000006	152	1521
17133	VASEN VASE 20 CLEAR GLASS	0,81818	7,07424	0,7603	VEN02000006	163	1633
26707	ENJE RLLER BLIND 120X250 WHITE	16,5207	48,2466	8,55	VEN02000006	121	1213
28508	LEN N PILLOW COT 35X55 WHITE	1,97521	30,55928	1,3716	VEN02000006	91	914
30196	IKEA 365+ SIDE PLATE 18X18 BEIGE	8,2562	100	0,98	VEN02000006	151	1511
30262	FORVAR JAR/LID 1,8 L GLASS/ALUMINIUM-COLOUR	1,2314	-10,77635	1,3641	VEN02000006	143	1432
30993	GRANAT N CUSHION 50X50 WHITE	8,2562	100	1,7899	VEN02000006	123	1231
31841	EDSVIK N KIT MX TP DUAL CHROME-PLATED	41,31405	41,47875	24,1775	VEN02000006	75	751
35127	HASTVEDA ACH RATTAN	47,6281	42,4568	27,4067	VEN02000006	12	125
37980	MELODI PEND LMP	4,12397	32,05576	2,802	VEN02000006	102	1023
39804	BRUNKRISSLA QC/2PWC 200X200/50X60 RED	14,86777	18,52241	12,1139	VEN02000006	111	1112
46786	SKYDD WOOD TREATMENT OIL INDOOR USE 0,5 L	4,12397	46,62425	2,2012	VEN02000006	142	1424
49520	TUPPLUR RLLER BLIND 140X195 WHITE	61,9752	100	13,8699	VEN02000006	121	1213
54564	CAPITA N LEG 16 STAINLESS STEEL 4-P	30,5785	100	0	VEN02000006	71	711
55691	LOCK CEIL LMP WHITE	3,29752	29,94432	2,3101	VEN02000006	102	1022
56884	GALANT FRAME 160 SILVER-COLOUR	32,2314	69,08853	9,9632	VEN02000006	31	312
57256	BLANDA BLANK SERV BOWL 36 STAINLESS STEEL	8,2562	35,31649	5,3404	VEN02000006	151	1511
58487	FLYTTA KITCHEN TROLLEY 98X57 STAINLESS STEEL	114,87603	46,81745	61,094	VEN02000006	78	782
58519	GORM SHELVT 78X55X174 SOFTWOOD	95,0413	100	20,8729	VEN02000006	191	1911
63672	TROFAST FRAME 44X91 PINE	19,8265	13,2724	17,195	VEN02000006	92	921

Рисунок 3.7 - Приклад таблиці вхідних даних товарів

### 3.4.2 Завантаження вхідних даних

Для завантаження вхідних даних необхідно скористатися пунктом меню «Завантажити дані» та по черзі обрати пункти «Транзакції», «Категорії товарів», «Групи продуктів» та «Товари», що зображені на рис. 3.8.

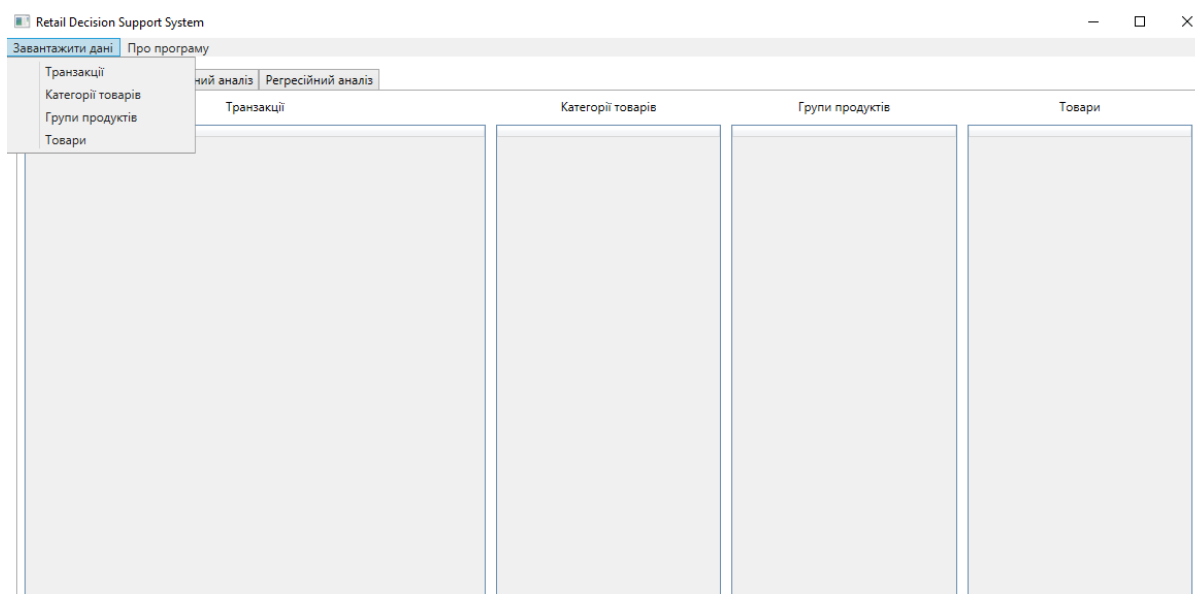


Рисунок 3.8 - Завантаження вхідних даних

Далі обирається файл з відповідними даними за зразком рис. 3.9.

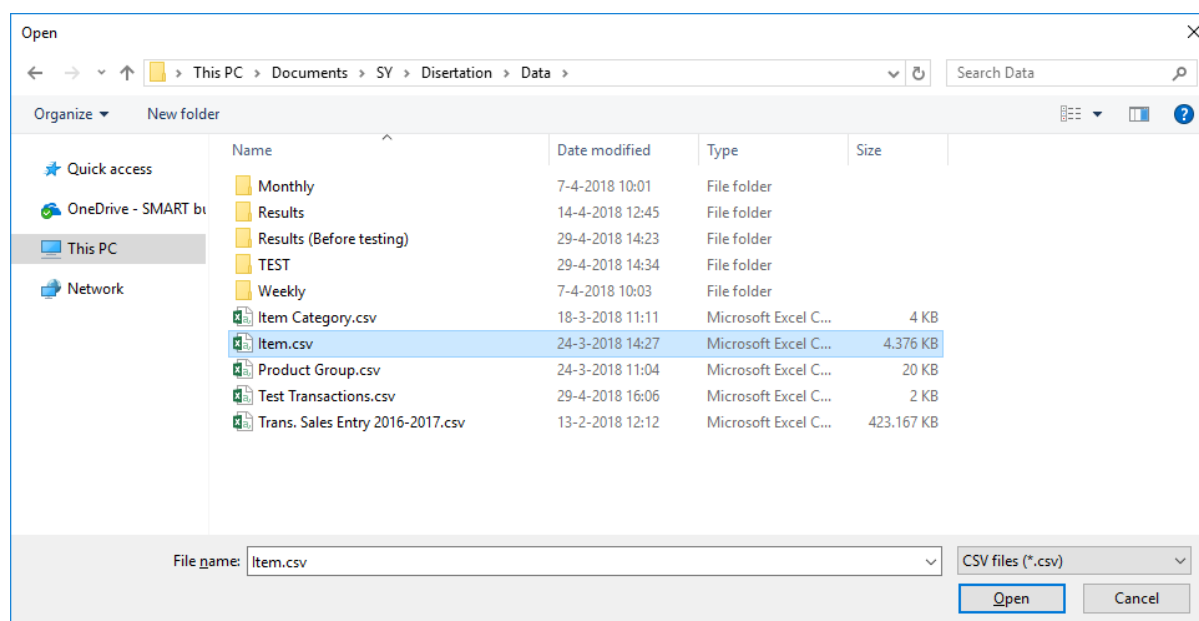


Рисунок 3.9 - Діалогове вікно завантаження даних

Після завантаження даних, їх опис можна переглянути на вкладці «Дані», як показано на рис. 3.10.

Retail Decision Support System

Завантажити дані Про програму

Дані ABC-аналіз Асоціативний аналіз Регресійний аналіз

Транзакції					Категорії товарів		Групи продуктів		Товари	
TransactionNo	LineNo	ItemNo	ItemCategoryCode	ProductGroupCode	Code	Description	Code	Description	No	Description
1010007738	10000	00011536	076	0762	142	Kitchen tools	218	Sideboard and cabinets	137	MEASURING KITCHEN
1010007738	20000	00133038	141	1414	143	Storing and washing	221	Closed storage	203	KITCHEN PLUMBING
1010007738	30000	00141401	094	0942	151	Dining and serving	222	Display cabinets	496	BATHROOM PLUMBING
1010007738	40000	00149459	142	1421	152	Glassware	223	Open shelving units & Book	528	INST IKEA GAS APP
1010007738	50000	00150004	111	1111	153	Cutlery	224	Drawer unit/cabinet series -	533	inst of wrktp
1010007738	60000	00152159	142	1422	154	Set the table	225	OFFICE FURNITURE	552	CONN UPPELVA TV DEV
1010007738	70000	00152282	082	0822	155	Coffee and tea.	226	CONFERENCE AND OFFICE	566	HOME DELIV OUTSD REGIC
1010007738	80000	00165213	093	0936	161	Vases and flowers	231	Tv furniture	571	LOAD SERV VAT SPEC
1010007738	90000	00177225	076	0761	162	Green decoration	232	Media electronics & access	11013	LACK WLL SHLF 190X26 WH
1010007738	100000	00201904	076	0762	163	Home decoration	241	Bookcases..	11428	GRUNDTAL WLL SHLF 80 ST
1010007738	110000	00204629	073	0731	164	Wall decoration.	242	Storage solutions.	11536	GRUNDTAL S-HOOK 11 STA
1010007738	120000	00215312	073	0731	165	Paper shop	243	Display/collect solutions	15134	SVALKA WHI WNE GLS 25 C
1010007738	130000	00238993	142	1421	166	Decoration objects	244	Entertainment solutions	17133	VASEN VASE 20 CLEAR GLA
1010007738	140000	00251550	163	1632	171	Outdoor furniture.	245	Interior fittings & accessorie	26707	ENJE RLLER BLIND 120X250
1010007738	150000	00258839	154	1542	172	Outdoor parasols and acces	251	Occasional tables	28508	LEN N PILLOW COT 35X55 V
1010007738	160000	00271874	112	1121	181	Small storage.	252	Furniture care & treatment	30196	IKEA 365+ SIDE PLATE 18X1
1010007738	170000	00275848	021	0215	182	Clothes and shoes organisa	261	Free-standing shelving unit:	30262	FORVAR JAR/LID 1,8 L GLAS
1010007738	180000	00288935	031	0311	183	Laundry, cleaning and sortir	311	Desk and tables for home	30993	GRANAT N CUSHION 50X50
1010007738	190000	00292895	022	0222	184	Clothes & shoe organisers &	312	Desks and tables for profes:	31841	EDSVIK N KIT MX TP DUAL
1010007738	200000	00298128	042	0423	185	Sorting solutions	313	Workspaces table bar	35127	HASTVEDA ACH RATTAN
1010007738	210000	00299646	113	1132	186	Cleaning & laundry	314	Workspaces mobile solution	37980	MELODI PEND LMP
1010007738	220000	00305614	012	0125	191	Secondary storage range	315	Workspace accessories	39804	BRUNKRISSLA QC/2PWC 20

Рисунок 3.10 - Вкладка опису вхідних даних

### 3.4.3 Попередній аналіз завантажених даних

Після завантаження даних, користувач має можливість провести ABC-аналіз на відповідній вкладці, обравши при цьому опцію «Категорії товарів», «Групи продуктів» чи «Товари», як це зображено на рис. 3.11 та опцію «По кількості» чи «По прибутку», як це зображено на рис. 3.12.

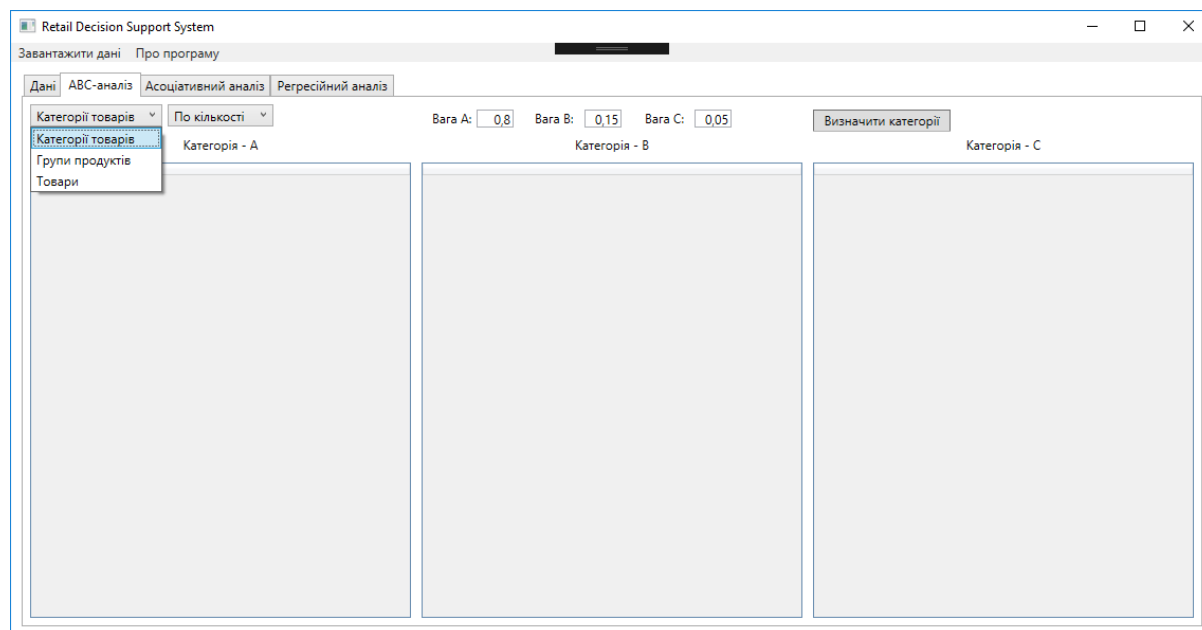


Рисунок 3.11 - Вкладка ABC-аналізу з опцією «Категорії товарів»

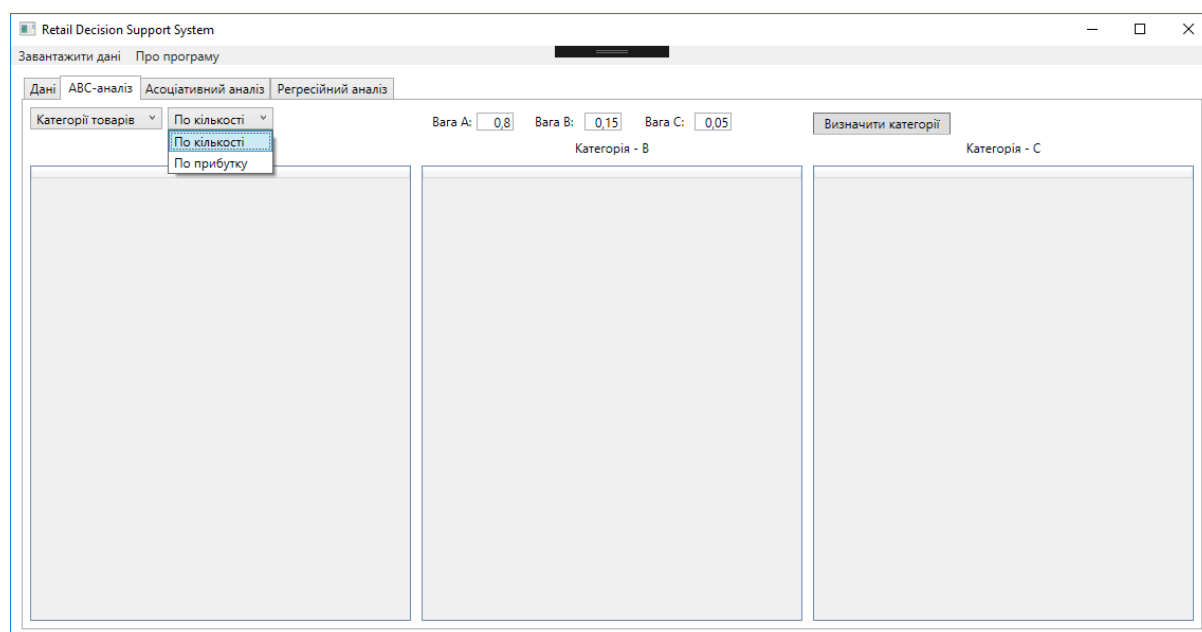


Рисунок 3.12 - Вкладка ABC-аналізу з опцією «По кількості»

Також потрібно вказати вагу кожної з категорій А, В та С та натиснути кнопку «Визначити категорії». Після цього буде згенеровано категорії відповідно до заданих ваг, як це зображено на рис. 3.13.

Retail Decision Support System

Завантажити дані Про програму

Дані ABC-аналіз Асоціативний аналіз Регресійний аналіз

Категорії товарів По кількості

Bara A: 0,8 Bara B: 0,15 Bara C: 0,05

Визначити категорії

Категорія - A Категорія - B Категорія - C

Code	Description
962	Restaurant.
151	Dining and serving
963	Bistro
163	Home decoration
152	Glassware
182	Clothes and shoes organisation
181	Small storage.
143	Storing and washing
142	Kitchen tools
961	Swedish Food Market
162	Green decoration
113	Bath textiles.
076	Kitchen accessories
155	Coffee and tea.
141	Cookware
062	Bathroom organisers
123	Cushions, throws and chairpads
941	Administrative Range 941
021	Multipurpose furniture
164	Wall decoration.
111	Bedlinen

Code	Description
103	Light sources
101	Luminaires, portables
112	Quilts and pillows
192	Transport and assembly range
082	Dining seating
131	Home furnishing rugs
073	Kitchen fixed interiors
105	Lighting accessories
183	Laundry, cleaning and sorting
171	Outdoor furniture.
031	Workspaces
191	Secondary storage range
922	Safety and security
921	Active life
072	Kitchen fronts
012	Armchairs, footstool & sofa ta
966	Other products.
071	Kitchen cabinets
102	Luminaires, fixtures
079	Knobs and handles

Code	Description
093	Children 3-7 years
043	Chests & other furniture
061	Bathroom furniture
165	Paper shop
051	Mattresses and accessories
153	Cutlery
122	Fabrics and accessories
033	Work seating range
041	Beds
044	Mirrors.
011	Sofas
078	Freestanding kitchens and kitc
075	Kitchen taps, sinks and sink a
924	Life at home knowledge and hom
095	Children 8-12 year
104	Integrated lighting
953	Administrative Range 953
081	Dining tables
108	Seasonal lighting
074	Kitchen worktops
022	Store and display furniture

Рисунок 3.13 - Вікно результатів ABC-аналізу

### 3.4.4 Вибір та налаштування параметрів для асоціативного аналізу

На вкладці «Асоціативний аналіз», необхідно обрати одну із опцій «Категорії товарів», «Групи продуктів» чи «Товари» та ввести значення для полів «Підтримка» та «Впевненість», як це зображено на рис. 3.14.

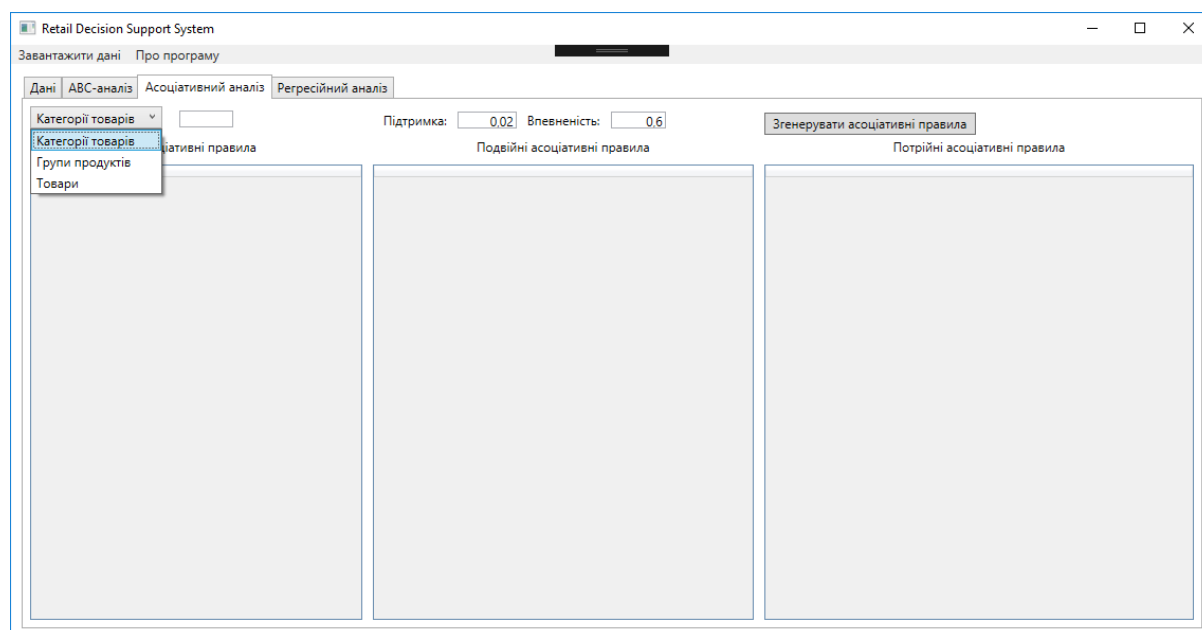


Рисунок 3.14 - Вкладка вибору параметрів Асоціативного аналізу

Після цього буде згенеровано і виведено асоціативні правила, як це зображено на рис. 3.15.

Rule	Support	Confidence
{ 123 } -> { 142 }	0.0415282392026578	0.4451
{ 154 } -> { 142 }	0.0357142857142857	0.8681
{ 091 } -> { 142 }	0.0465116279069767	0.7671
{ 121 } -> { 142 }	0.0249169435215947	0.6111
{ 094 } -> { 142 }	0.0514950166112957	0.7671
{ 121 } -> { 162 }	0.0257475083056478	0.4641
{ 113 } -> { 062 }	0.0456810631229236	0.5401
{ 113 } -> { 111 }	0.0440199335548173	0.75
{ 121 } -> { 113 }	0.0274086378737542	0.4921
{ 121 } -> { 076 }	0.0240863787375415	0.6311
{ 155 } -> { 141 }	0.0448504983388704	0.8
{ 062 } -> { 141 }	0.0548172757475083	0.4441
{ 123 } -> { 141 }	0.0406976744186047	0.7271
{ 154 } -> { 141 }	0.0390365448504983	0.48
{ 121 } -> { 141 }	0.0240863787375415	0.8571
{ 094 } -> { 141 }	0.042358803986711	0.5101
{ 121 } -> { 062 }	0.0240863787375415	0.6571
{ 123 } -> { 111 }	0.0406976744186047	0.7811
{ 121 } -> { 123 }	0.0240863787375415	0.4621

Рисунок 3.15 - Відображення згенерованих асоціативних правил



### 3.4.5 Вибір та налаштування параметрів для регресійного аналізу

На вкладці «Регресійний аналіз» для завантаження підготовлених статистичних даних, необхідно натиснути на кнопку «Завантажити», після вибору файлу з даними, вони відобразяться у вікні, як це зображено на рис. 3.16.

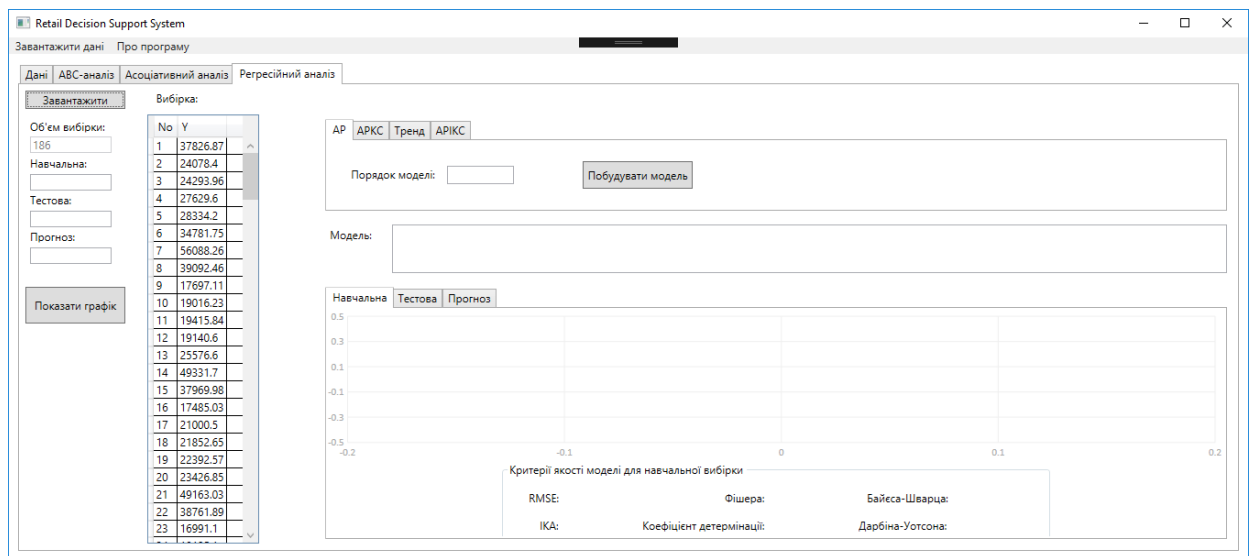


Рисунок 3.16 – Вкладка «Регресійний аналіз»

Після цього необхідно ввести розмір навчальної, тестової вибірок та прогноз. Оглянути вибірку на графіку можна за допомогою кнопки «Показати графік», що зображено на рис. 3.17.

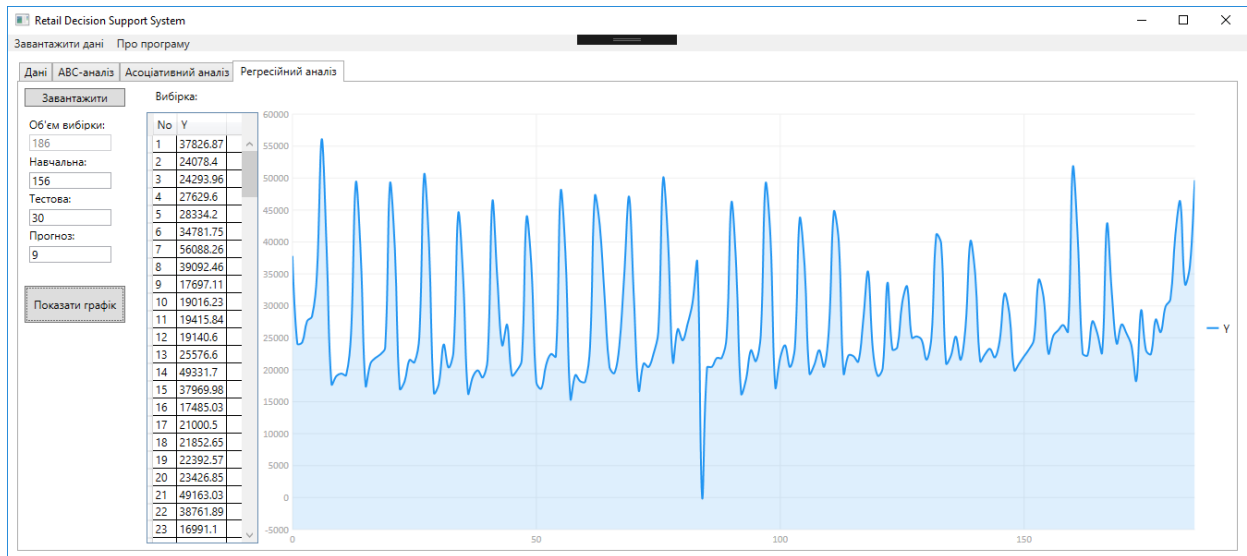


Рисунок 3.17 – Приклад графіку завантажених даних

Для побудови моделі, необхідно ввести налаштування та натиснути кнопку «Побудувати модель», програма відобразить модель, побудовані графіки на оригінальних, навчальних та тестових даних і графік прогнозу, а також обчислені оцінки якості моделі, як це зображено на рисунках 3.18 – 3.20.

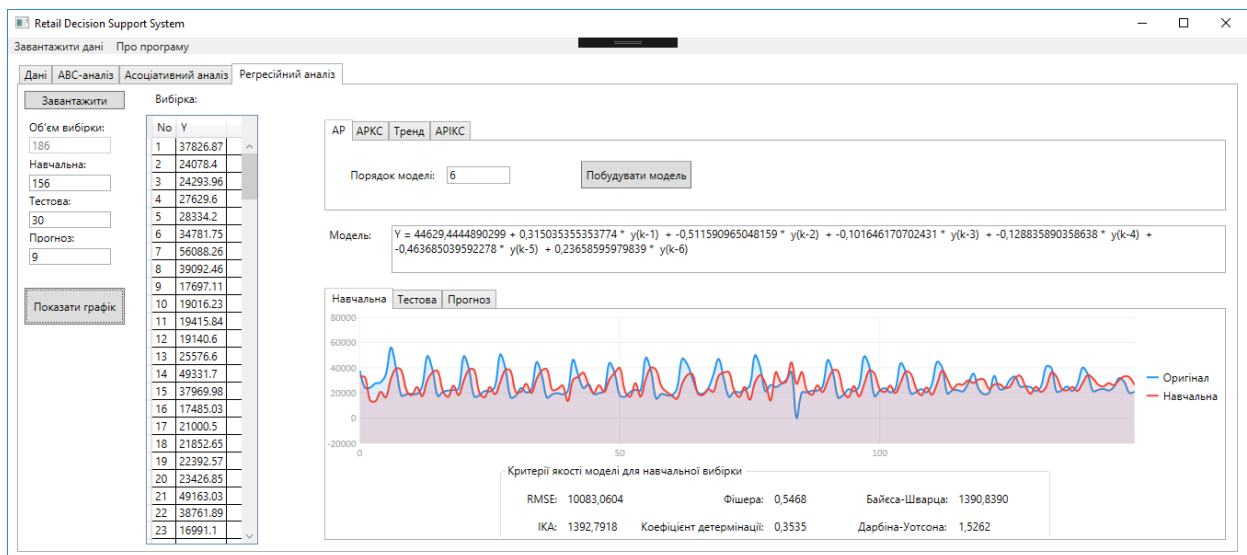


Рисунок 3.18 – Приклад виведення результатів для навчальних даних

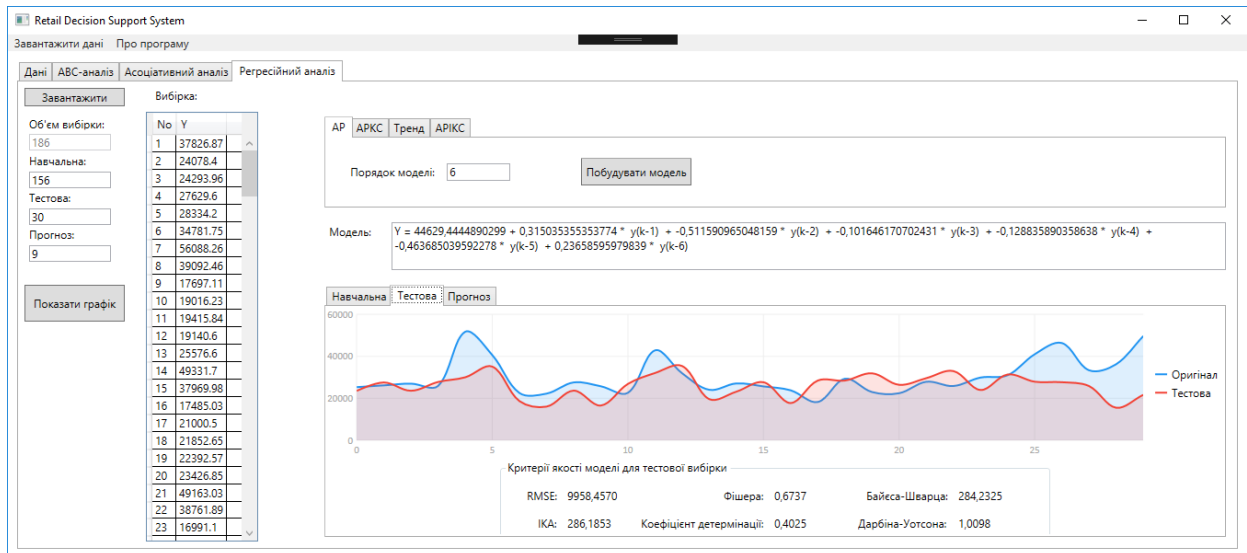


Рисунок 3.19 – Приклад виведення результатів для тестових даних

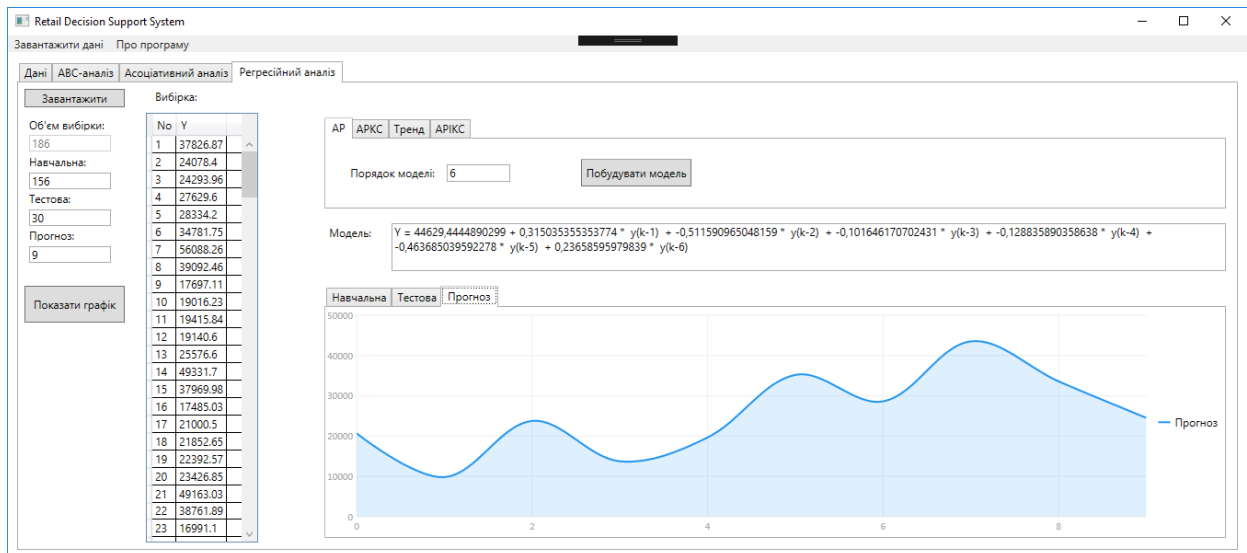


Рисунок 3.20 – Приклад виведення прогнозу

### 3.5 ABC-аналіз категорій товарів

Програму застосовано на прикладі даних меблевого магазину, що містять інформацію про категорії товарів (загальною кількістю 156 найменувань), групи продуктів (загальною кількістю 709 найменувань),

товари (загальною кількістю 19716 найменувань), а також інформацію про транзакції за період з 2016 року загальною кількістю 3352667 операцій.

Проведено АВС-аналіз категорій товарів, по показав найбільш актуальні категорії. Результат проведеного АВС-аналізу по категоріям, а саме сформовану А-категорію наведено у таблиці 3.1.

Таблиця 3.1 – А-категорія, АВС-аналізу по категоріям товару

Код категорії товару	Опис категорії товару	Сумарний прибуток по всіх транзакціях
21	Multipurpose furniture	269553,186
962	Restaurant.	239108,595
42	Wardrobes.	233487,112
11	Sofas	212956,0236
111	Bedlinen	210062,5595
43	Chests & other furniture	185184,8007
51	Mattresses and accessories	165022,2944
82	Dining seating	164386,2251
61	Bathroom furniture	157680,538
171	Outdoor furniture.	151260,2346
31	Workspaces	145241,8478
12	Armchairs, footstool & sofa ta	141269,3429
41	Beds	139906,4103
182	Clothes and shoes organisation	133654,9586
131	Home furnishing rugs	109534,1495
112	Quilts and pillows	103344,7212
121	Window solutions.	101878,3876
72	Kitchen fronts	96953,53034
77	Kitchen appliances	96447,23
73	Kitchen fixed interiors	94584,0918
141	Cookware	93864,7974
163	Home decoration	92471,2048
94	Play	88498,15438
101	Luminaires, portables	88206,56408
181	Small storage.	86150,0717
33	Work seating range	85485,2136
81	Dining tables	81188,4014

91	Baby	78916,40407
142	Kitchen tools	78298,86403
102	Luminaires, fixtures	75058,17586

### Продовження таблиці 3.1

162	Green decoration	74726,1074
151	Dining and serving	74423,9624
113	Bath textiles.	74293,2577
92	Childrens storage	73179,3265
123	Cushions, throws and chairpads	70390,5793
71	Kitchen cabinets	69560,4348

Відповідно до отриманих результатів, найбільш цікавими для аналізу є категорії товарів, що займають перші позиції в А-категорії: Багатофункціональні меблі («Multipurpose furniture»), Ресторан («Restaurant») та Шафи («Wardrobes»). Оскільки саме ці категорії приносять найбільшу частку прибутку магазину.

### 3.6 Побудова та аналіз асоціативних правил

Отримані асоціативні правила для категорій товарів, довжиною 2 із підтримкою більше 0.02 і впевненістю 0.3, зображені у таблиці 3.2.

Таблиця 3.2 – Асоціативні правила довжиною 2 для категорій товарів

Правило	Підтримка	Впевненість	Поліпшення
{ 141 } → { 142 }	0,029	0,4284	4,8601
{ 143 } → { 142 }	0,0312	0,3719	4,2193
{ 151 } → { 142 }	0,0201	0,3697	4,1936

$\{ 141 \} \rightarrow \{ 143 \}$	0,0247	0,3657	4,364
$\{ 062 \} \rightarrow \{ 182 \}$	0,0218	0,3569	3,6346
$\{ 142 \} \rightarrow \{ 143 \}$	0,0312	0,3536	4,2193
$\{ 152 \} \rightarrow \{ 142 \}$	0,0212	0,3462	3,9269
$\{ 143 \} \rightarrow \{ 182 \}$	0,0278	0,3315	3,3764
$\{ 141 \} \rightarrow \{ 182 \}$	0,0224	0,3309	3,3696

Продовження таблиці 3.2

$\{ 941 \} \rightarrow \{ 163 \}$	0,0261	0,3297	3,6864
$\{ 142 \} \rightarrow \{ 141 \}$	0,029	0,3287	4,8601
$\{ 142 \} \rightarrow \{ 182 \}$	0,0287	0,3258	3,3176
$\{ 941 \} \rightarrow \{ 182 \}$	0,0255	0,3223	3,2826
$\{ 941 \} \rightarrow \{ 142 \}$	0,0252	0,3187	3,6158
$\{ 941 \} \rightarrow \{ 143 \}$	0,0251	0,3165	3,7768

Отримані асоціативні правила для категорій товарів, довжиною 3 із підтримкою більше 0.012 і впевненістю більшою 0.2, зображені у таблиці 3.3. Де ми можемо побачити, що згенеровані правила мають меншу підтримку та впевненість.

Таблиця 3.3 – Асоціативні правила довжиною 3 для категорій товарів

Правило	Підтримка	Впевненість	Поліпшення
{ 962 151 } → { 182 }	0,0121	0,4399	4,4797
{ 962 151 } → { 152 }	0,0133	0,4272	6,9658
{ 962 151 } → { 163 }	0,014	0,4114	4,6005
{ 962 152 } → { 151 }	0,0133	0,3678	6,7564
{ 962 163 } → { 151 }	0,014	0,2515	4,6185
{ 962 182 } → { 151 }	0,0121	0,2424	4,4517

Отримані асоціативні правила для категорій товарів, довжиною 4 із підтримкою більше 0.005 і впевненістю 0.55, зображені у таблиці 3.4.

Таблиця 3.4 – Асоціативні правила довжиною 3 для категорій товарів

Правило	Підтримка	Впевненість	Поліпшення
{ 151 143 141 } → { 142 }	0,0058	0,6846	7,7661
{ 152 143 141 } → { 142 }	0,0055	0,6778	7,6891
{ 143 141 062 } → { 142 }	0,0051	0,676	7,6686
{ 182 143 141 } → { 142 }	0,007	0,6499	7,3724

Продовження таблиці 3.4

{ 163 143 141 } → { 142 }	0,0056	0,6252	7,0925
{ 142 141 062 } → { 143 }	0,0051	0,6051	7,2205
{ 151 142 141 } → { 143 }	0,0058	0,5906	7,0483
{ 152 142 141 } → { 143 }	0,0055	0,588	7,017
{ 143 141 941 } → { 142 }	0,0052	0,5879	6,6699

{ 151 143 142 } → { 141 }	0,0058	0,5829	8,6172
{ 182 142 141 } → { 143 }	0,007	0,5794	6,9138
{ 182 143 062 } → { 142 }	0,0051	0,5781	6,5581
{ 182 142 062 } → { 143 }	0,0051	0,5619	6,7055
{ 163 142 141 } → { 143 }	0,0056	0,5603	6,6868
{ 143 142 062 } → { 141 }	0,0051	0,5552	8,2085
{ 143 142 062 } → { 182 }	0,0051	0,5525	5,6263

Як ми можемо побачити, дані правила володіють значно меншою підтримкою, оскільки не так часто зустрічаються у транзакціях, але вони мають велику впевненість, що означає те, що товари з вищезгаданих категорій товарів частіше купують разом.

Відповідно до наведених таблиць, можемо зробити висновки, що покупці зазвичай при покупці кухонного посуду, миючих засобів також купують кухонне приладдя. А при покупці полицок для ванної кімнати, разом купують і полиці для взуття та одягу.

### 3.7 Побудова моделей ключових показників роздрібного магазину

#### 3.7.1 Опис вихідних даних

Для моделювання та прогнозування основних показників, транзакції роздрібного підприємства було просумовано по днях. Що дозволило



отримано ряди статистичних даних для прогнозування прибутків підприємства, завантаженості магазину (загальна кількість транзакцій) та попиту на основні категорії товарів, що приносять найбільші прибутки відповідно до проведеного ABC-аналізу.

### 3.7.2 Опис та аналіз отриманих моделей

Отримані моделі для завантаженості магазину:

1. AP(7):

$$Y = 46944,37 + 0,34 * y(k - 1) - 0,57 * y(k - 2) - 0,1 * y(k - 3) - 0,14 * y(k - 4) - 0,52 * y(k - 5) + 0,28 * y(k - 6) + 0,53 * y(k - 7)$$

Графік побудованої моделі зображено на рис. 3.21.

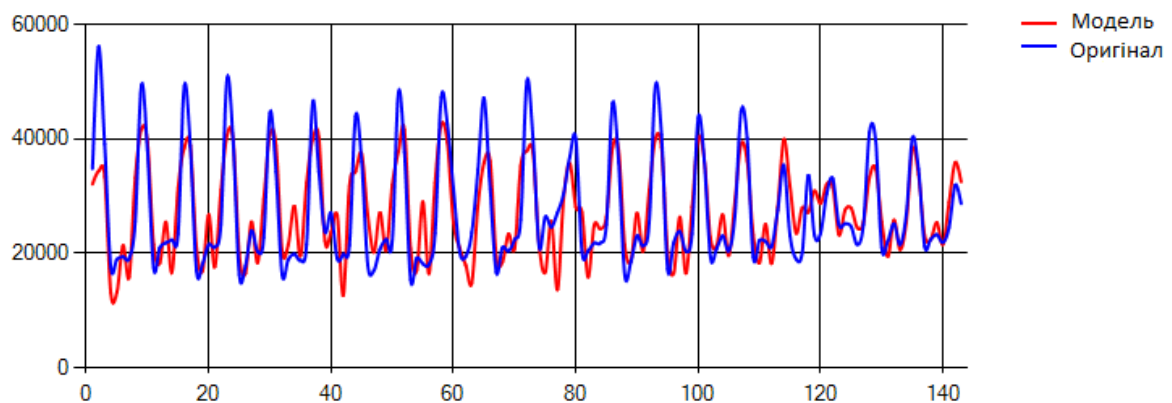


Рисунок 3.21 – AP(7) завантаженості на навчальній вибірці

2. АРКС(7,6):

$$\begin{aligned}
 Y = & 46944,37 + 0,34 * y(k-1) - 0,57 * y(k-2) - 0,09 * y(k-3) - \\
 & - 0,14 * y(k-4) - 0,52 * y(k-5) + 0,28 * y(k-6) + \\
 & + 0,53 * y(k-7) - 40,78 - 1,09 * ma(k-1) + 3,22 * ma(k-2) - \\
 & - 3,31 * ma(k-3) + 1,32 * ma(k-4) + 0,46 * ma(k-5) - \\
 & - 0,54 * ma(k-6)
 \end{aligned}$$

Графік побудованої моделі зображено на рис. 3.22.

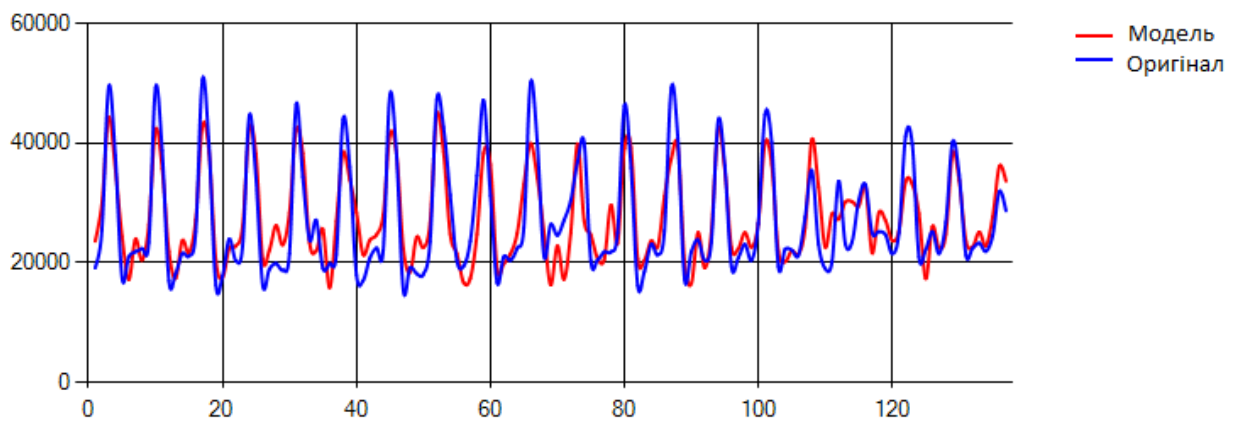


Рисунок 3.22 – АРКС(7,6) завантаженості на навчальній вибірці

3. АРІКС(7,6,2):

$$\begin{aligned}
 Y = & 1661,43 - 0,39 * \tilde{y}(k-1) - 0,42 * \tilde{y}(k-2) - 0,4 * \tilde{y}(k-3) - \\
 & - 0,41 * \tilde{y}(k-4) - 0,42 * \tilde{y}(k-5) - 0,39 * \tilde{y}(k-6) + 0,53 * \tilde{y}(k-7) \\
 & - 111,53 - 0,30 * ma(k-1) + 2,48 * ma(k-2) - \\
 & - 1,75 * ma(k-3) + 1,26 * ma(k-4) + 0,7 * ma(k-5) + 0,05 \\
 & * ma(k-6) + y(k) - 2 * y(k-1) + y(k-2)
 \end{aligned}$$

Графік побудованої моделі зображено на рис. 3.23.

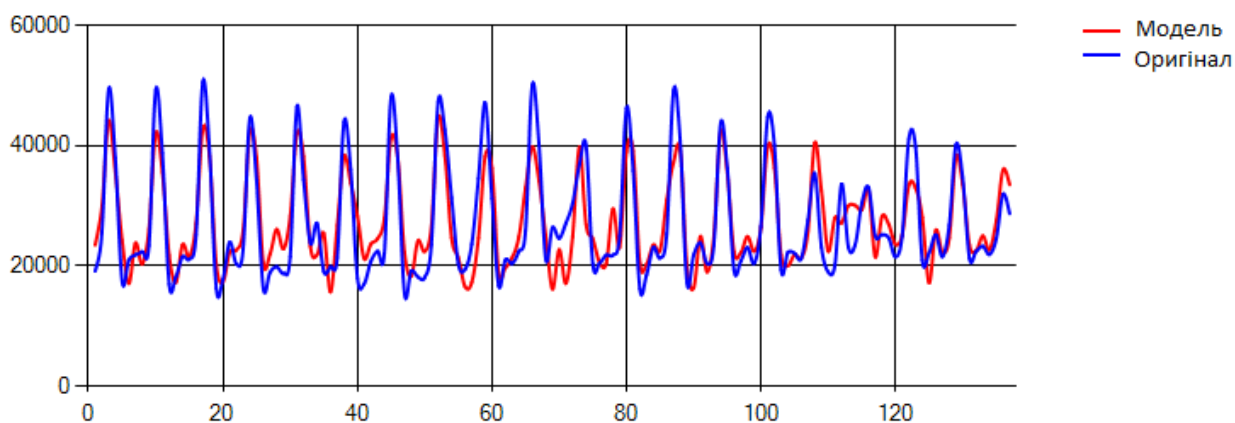


Рисунок 3.23 – АРІКС(7,6,2) завантаженості на навчальній вибірці

4. Тренд порядку 2:

$$Y(k) = 30555,48 - 79,82 * k + 0,42 * k^2$$

Графік побудованої моделі зображено на рис. 3.24.

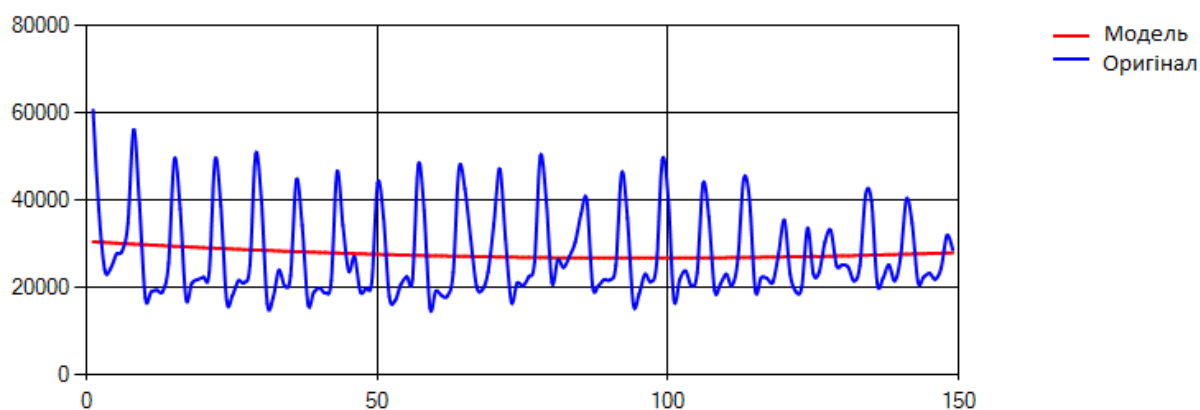


Рисунок 3.24 – Тренд(2) завантаженості на навчальній вибірці

Обраховані оцінки для побудованих моделей наведені в таблицях 3.5-3.6.

Таблиця 3.5 – Оцінки моделі завантаженості для навчальної вибірки

Модель	RMSE	Коефіцієнт детермінації	ІКА	Байєса- Шварца	
AP(7)	5614,54	0,66	1244,61	1242,66	
APKC(7,6)	4691,22	0,63	1168,36	1166,4	
APIKC(7,6,2)	4691,96	0,63	1168,17	1166,22	
Тренд(2)	9811,14	0,01	1379,53	1377,58	

Таблиця 3.6 – Оцінки моделі завантаженості для тестової вибірки

Модель	RMSE	Коефіцієнт детермінації	ІКА	Байєса- Шварца	
AP(7)	8041,47	0,56	342,72	340,77	1
APKC(7,6)	9305,13	0,75	348,11	346,16	3
APIKC(7,6,2)	9322,99	0,75	348,2	346,24	3
Тренд(2)	7431,63	0,01	339,8	337,84	0

Відповідно до обрахованих оцінок моделей, найкращою моделлю є APIKC(7,6,2).

Отримані моделі для прибутку магазину:

1. AP(7):

$$Y = 75623,15 + 0,05 * y(k - 1) - 0,59 * y(k - 2) - \\ - 0,24 * y(k - 3) - 0,29 * y(k - 4) - 0,55 * y(k - 5) + \\ + 0,02 * y(k - 6) + 0,6 * y(k - 7)$$

Графік побудованої моделі зображено на рис. 3.25.

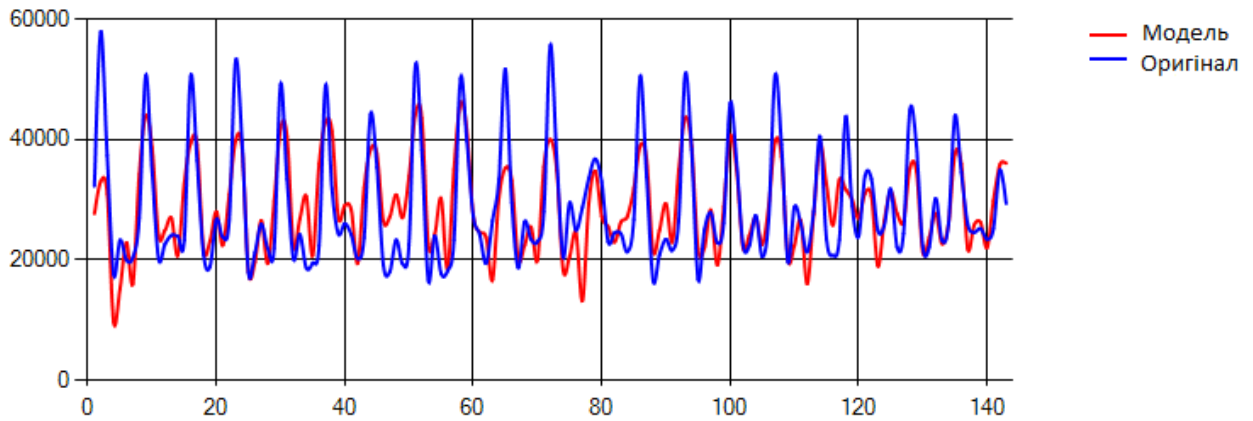


Рисунок 3.25 – AR(7) прибутковості на навчальній вибірці

2. АРКС(7,4):

$$\begin{aligned}
 Y = & 75623,15 + 0,05 * y(k - 1) - 0,59 * y(k - 2) - \\
 & - 0,24 * y(k - 3) - 0,29 * y(k - 4) - 0,55 * y(k - 5) + \\
 & + 0,01 * y(k - 6) + 0,6 * y(k - 7) - 95,81 - 0,04 * ma(k - 1) + \\
 & + 1,1 * ma(k - 2) - 1,34 * ma(k - 3) + 0,78 * ma(k - 4)
 \end{aligned}$$

Графік побудованої моделі зображено на рис. 3.26.

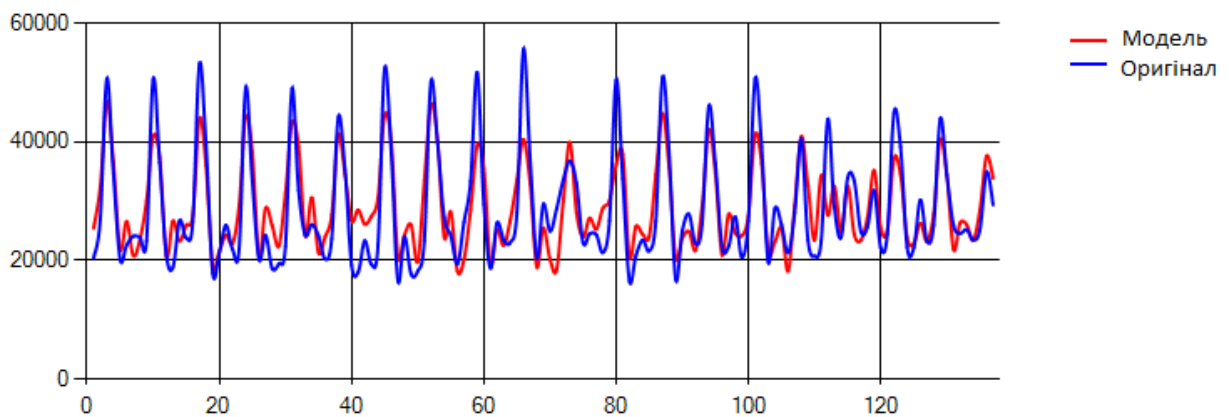


Рисунок 3.26 – АРКС(7,4) прибутковості на навчальній вибірці

3. АРІКС(7,4,2):

$$\begin{aligned}
 Y = & 2676,67 - 0,4 * \tilde{y}(k - 1) - 0,42 * \tilde{y}(k - 2) - \\
 & - 0,41 * \tilde{y}(k - 3) - 0,41 * \tilde{y}(k - 4) - 0,42 * \tilde{y}(k - 5) - \\
 & - 0,4 * \tilde{y}(k - 6) + 0,6 * \tilde{y}(k - 7) - 122,34 + 0,38 * ma(k - 1) + \\
 & + 1,11 * ma(k - 2) - 0,47 * ma(k - 3) + 0,91 * ma(k - 4) + \\
 & + y(k) - 2 * y(k - 1) + y(k - 2)
 \end{aligned}$$

Графік побудованої моделі зображено на рис. 3.27.

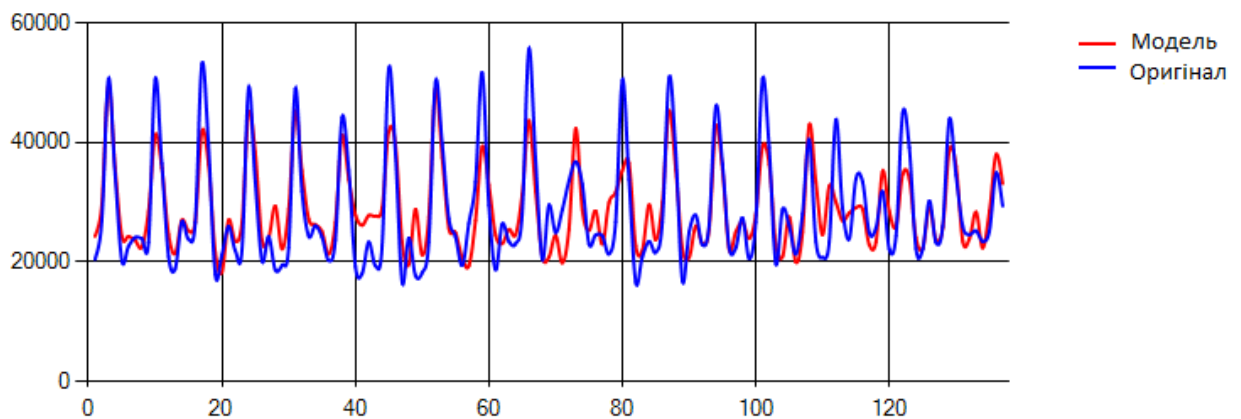


Рисунок 3.27 – АРІКС(7,4,2) прибутковості на навчальній вибірці

4. Тренд порядку 4:

$$Y(k) = 37867,89 - 723,92 * k + 16,25 * k^2 - 0,14 * k^3 + 0,0004 * k^4$$

Графік побудованої моделі зображено на рис. 3.28.

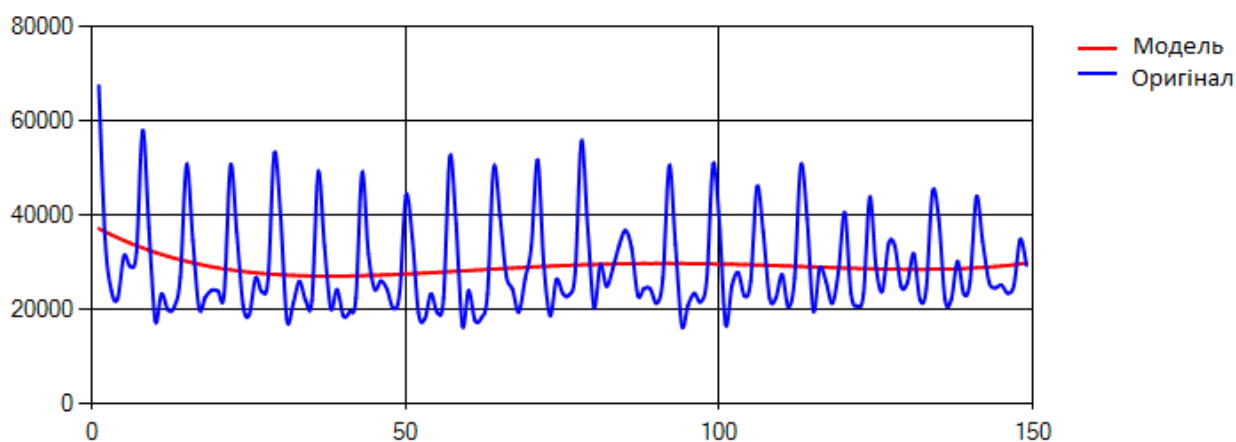


Рисунок 3.28 – Тренд(4) прибутковості на навчальній вибірці

Обраховані оцінки для побудованих моделей наведені в таблицях 3.7-3.8.

Таблиця 3.7 – Оцінки моделі прибутковості для навчальної вибірки

Модель	RMSE	Коефіцієнт детермінації	ІКА	Байеса- Шварца	
AP(7)	6231,88	0,59	1259,55	1257,59	
АРКС(7,4)	5090,52	0,54	1179,5	1177,55	
АРІКС(7,4,2)	5091	0,54	1179,33	1177,37	
Тренд(4)	9931,41	0,03	1381,05	1379,09	

Таблиця 3.8 – Оцінки моделі прибутковості для тестової вибірки

Модель	RMSE	Коефіцієнт детермінації	ІКА	Байеса- Шварца	
AP(7)	117,81	0,73	356,61	354,66	2
АРКС(7,4)	1312,23	0,92	360,92	358,97	1
АРІКС(7,4,2)	1322,94	0,92	361,1	359,15	1
Тренд(4)	1101,76	0,63	354,36	352,41	1

Відповідно до обрахованих оцінок моделей, найкращою моделлю є АРКС(7,4).

Отримані моделі попиту на категорію товарів 21 «Multipurpose furniture»:

1. AP(7):

$$Y = 257,68 + 0,110 * y(k - 1) - 0,13 * y(k - 2) - 0,01 * y(k - 3) - 0,04 * y(k - 4) - 0,15 * y(k - 5) + 0,04 * y(k - 6) + 0,59 * y(k - 7)$$

Графік побудованої моделі зображено на рис. 3.29.

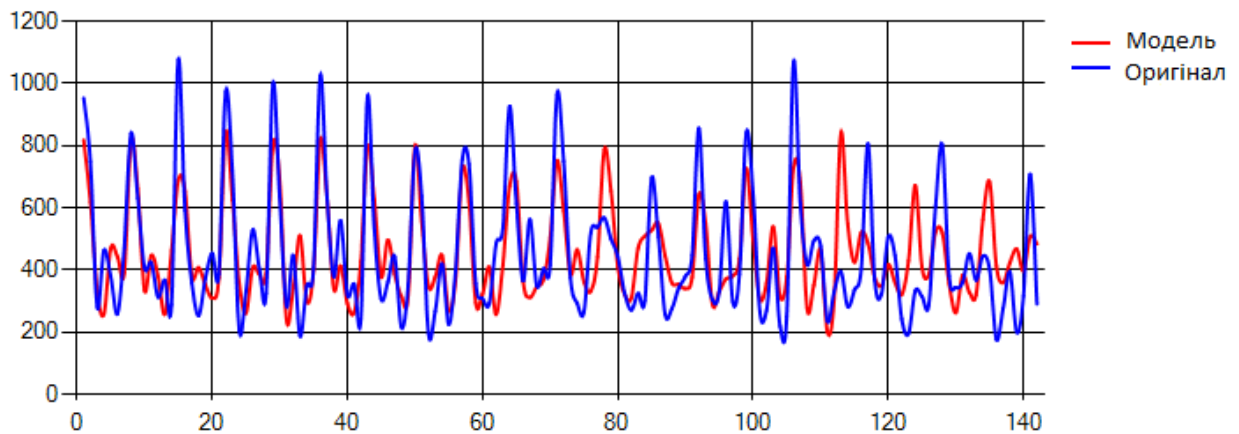


Рисунок 3.29 – AP(7) попиту на навчальній вибірці

2. APКС(7,2):

$$Y = 257,68 + 0,11 * y(k - 1) - 0,13 * y(k - 2) - 0,01 * y(k - 3) - 0,04 * y(k - 4) - 0,15 * y(k - 5) + 0,04 * y(k - 6) + 0,6 * y(k - 7) + 5,82 + 0,08 * ma(k - 1) - 0,09 * ma(k - 2)$$

Графік побудованої моделі зображено на рис. 3.30.



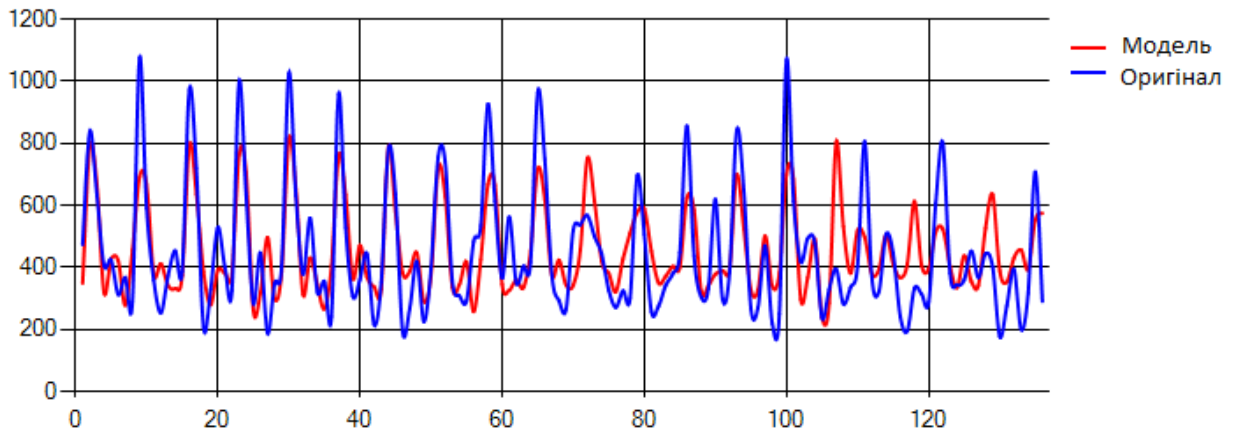


Рисунок 3.30 – АРКС(7,2) попиту на навчальній вибірці

3. АРІКС(7,2,2):

$$\begin{aligned}
 Y = & 8,72 - 0,39 * \tilde{y}(k - 1) - 0,4 * \tilde{y}(k - 2) - 0,4 * \tilde{y}(k - 3) - \\
 & - 0,4 * \tilde{y}(k - 4) - 0,4 * \tilde{y}(k - 5) - 0,39 * \tilde{y}(k - 6) - 0,38 * \tilde{y}(k - 7) + \\
 & + 2,34 + 0,45 * ma(k - 1) + 0,34 * ma(k - 2) + y(k) - \\
 & - 2 * y(k - 1) + y(k - 2)
 \end{aligned}$$

Графік побудованої моделі зображено на рис. 3.31.

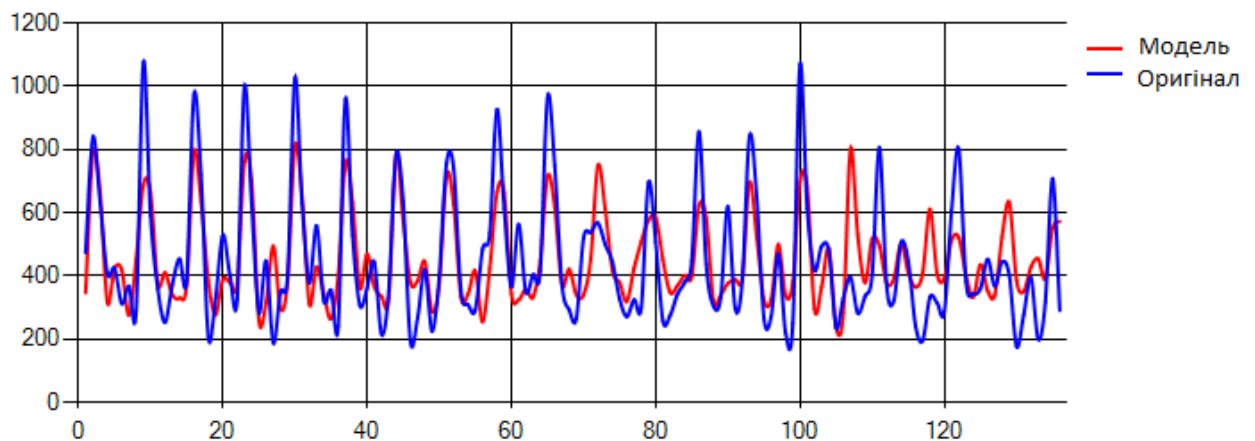


Рисунок 3.31 – АРІКС(7,2,2) попиту на навчальній вибірці

4. Тренд порядку 4:

$$Y(k) = 593,01 - 7,58 * k + 0,16 * k^2 - 0,001 * k^3 + 0,004 * k^4$$

Графік побудованої моделі зображено на рис. 3.32.

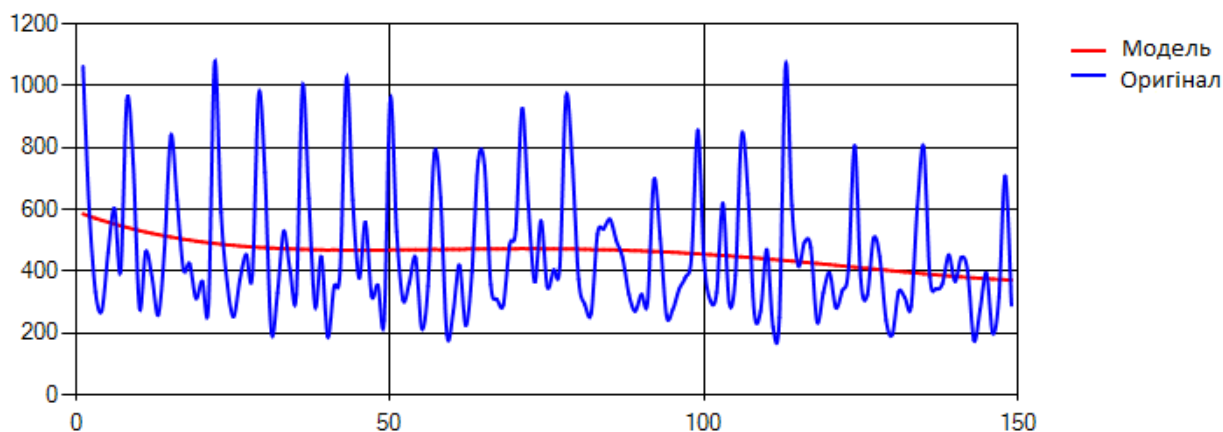


Рисунок 3.31 – Тренд(4) попиту на навчальній вибірці

Обраховані оцінки для побудованих моделей наведені в таблицях 3.9-3.10.

Таблиця 3.9 – Оцінки моделі попиту для навчальної вибірки

Модель	RMSE	Коефіцієнт детермінації	ІКА	Байєса- Шварца	Ф
АР(7)	141,53	0,54	713,26	711,3	1,
АРКС(7,2)	138,61	0,48	680,7	678,75	0,
АРІКС(7,2,2)	138,55	0,48	680,65	678,69	0,
Тренд(4)	206,92	0,04	804,52	802,56	0,

Таблиця 3.10 – Оцінки моделі попиту для тестової вибірки

Модель	RMSE	Коефіцієнт детермінації	ІКА	Байєса- Шварца	Ф
AP(7)	154,12	0,44	196,4	194,44	0,
АРКС(7,2)	155,8	0,51	196,8	194,85	1,
АРІКС(7,2,2)	155,99	0,51	196,84	194,89	1,
Тренд(4)	153,81	0,03	196,32	194,37	0,

Відповідно до обрахованих оцінок моделей, найкращою моделлю є АРІКС(7,2,2).

#### Висновки до розділу

В даному розділі було розглянуто архітектуру СППР, описані функціональні блоки та інтерфейс програмного продукту.

В програмі реалізовані наступні методи: АВС-аналіз, алгоритм Apriori для пошуку асоціативних правил, модель авторегресії, модель авторегресії з ковзним середнім, модель поліноміального тренду та модель авторегресії з інтегрованим ковзним середнім.

Програму застосовано на прикладі даних меблевого ринку, а саме проведений АВС-аналіз та побудовані асоціативні правила для категорій товарів. Побудовано моделі прогнозу попиту основних категорій товарів, прибутку та загальної завантаженості магазину, проведено їх аналіз та визначено найкращу.

## РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

### 4.1 Опис ідеї проекту

Опис ідеї стартап проекту описано в таблиці 4.1.

Таблиця 4.1 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення системи підтримки прийняття рішень для підприємств роздрібної торгівлі на основі методів інтелектуального аналізу даних	Роздрібна торгівля	Збільшення прибутків підприємства
	Управління запасами	Вчасна закупка актуальних товарів

Сильні, слабкі та нейтральні характеристики ідеї проекту зображено в таблиці 4.2.

Таблиця 4.2 - Визначення характеристик ідеї проекту

п/п	Техніко-економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтр. сторона)	S (сильна сторона)
		FS	ales oft	P lan. Analyt.			
.	Універсальний формат даних			-	+		
.	ABC-аналіз			-			+

.	Асоц. аналіз			+			+
.	Моделюва ння і прогнозування			+		+	
.	Відомість бренду			-	+		

#### 4.2 Технологічний аудит ідеї проекту

Технологічний аудит ідеї проекту наведений у таблиці 4.3.

Таблиця 4.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Попередній АВС-аналіз	C#, .NET	Технологія наявна і не потребує змін. Потрібно реалізувати алгоритм.	Технологія загальнодоступна
2	Асоціативний аналіз	C#, .NET	Необхідно реалізувати алгоритм Apriori.	Технологія загальнодоступна
3	Моделювання та прогнозування	C#, .NET	Необхідно реалізувати моделі: AR, ARMA, ARIMA	Технологія загальнодоступна
Обрана технологія реалізації ідеї проекту: для реалізації проекту обрана мова програмування C#, на основі технології .NET.				

### 4.3 Аналіз ринкових можливостей запуску стартап-проекту

Характеристика потенційного ринку стартап-проекту наведена у таблиці 4.4.

Таблиця 4.4 - Попередня характеристика потенційного ринку стартап-проекту

п/п	Показники стану ринку (найменування)	Характеристика
	Кількість головних гравців, од	4
	Загальний обсяг продаж, грн/ум.од	1000 ум.од
	Динаміка ринку (якісна оцінка)	Зростає
	Наявність обмежень для входу (вказати характер обмежень)	Немає
	Специфічні вимоги до стандартизації та сертифікації	Немає
	Середня норма рентабельності в галузі (або по ринку), %	20%

Характеристика потенційних клієнтів стартап-проекту наведена в таблиці 4.5.

Таблиця 4.5 - Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
----------	-----------------------------	--	---	-----------------------------------

1	Прогнозування та передбачення потреб користувачів у роздрібній торгівлі	Середній та великий бізнес, що застосовують системи керування ресурсів підприємства	ERP система підприємства, розміри оброблюваних даних, технічні обмеження	Ефективність прогнозування Швидка обробка даних Оптимальне використання ресурсів
---	--	--	--	--

Можливі загрози для стартап-проекту наведені у таблиці 4.6.

Таблиця 4.6 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Невпорядкованість і неповнота даних	Клієнтські бази можуть містити невпорядковані дані і також певні дані можуть бути відсутніми	Додавання модуля попередньої обробки даних
2	Нестача технічних ресурсів	Клієнти можуть мати обмежені локальні технічні ресурси, недостатні для повноцінної роботи системи	Винесення модуля обчислення на сервери компаній-партнерів

Фактори можливостей наведені у таблиці 4.7.

Таблиця 4.7 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Хмарні обчислення	Можливість виконання усіх обчислень на віддалених серверах	Пристосування модулів обчислення для роботи на сервері
2	Коригування прогнозу	Можливість коригування прогнозу в режимі реального часу на основі власної бази даних та спорідненої інформації з інтернету	Розробка модулів інтеграції з google analytics, оптимізація застосунку для можливості роботи в реальному часі

Проведений ступеневий аналіз конкуренції на ринку зображено у таблиці 4.8.

Таблиця 4.8 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - олігополія	Присутня невелика кількість фірм. Більшість ринку контролюють фірми-гіганти	Впровадження технологічних інновацій. Кооперація з дослідницькими центрами.
2. За рівнем конкурентної боротьби - глобальний	Продукція не залежить від країни чи локалізації клієнта	Розширення функціоналу та задоволення потреб клієнтів.



3. За галузевою ознакою внутрішньогалузева	Продукт спрямований на роздрібну торгівлю	
4. Конкуренція за видами товарів: - за бажанням	Полягає у випередженні задоволення бажань клієнта	
5. За характером конкурентних переваг - нецінова	Переваги передбачають собою ефективність та різноманіття функціоналу	
6. За інтенсивністю - не марочна	Торгова марка майже немає впливу	

Проведений аналіз конкуренції в галузі зображено у таблиці 4.9.

Таблиця 4.9 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти
	SalesLoft, IBM, Planning Analytics	SlickPie
Висновки	Контролюють велику частину ринку, мають узагальнені рішення	Спрямовані на малий бізнес, не мають локалізацій для більшості країн Європи

Фактори конкурентоспроможності та їх обґрунтування наведені в таблиці 4.10.

Таблиця 4.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Інновації	Інноваційні рішення мають забезпечити перевагу нашим клієнтам над конкурентами
2	Функціонал	Функціонал повинен покривати вирішення необхідних задач клієнтів
3	Цінова політика	Вартість продукту відіграє велику роль при виборі системи клієнтом
4	Ресурсоємність	Великі затрати технічних ресурсів можуть спровокувати необхідність залучення додаткових коштів

Порівняльний аналіз сильних та слабких сторін проекту відображено у таблиці 4.11.

Таблиця 4.11 - Порівняльний аналіз сильних та слабких сторін RFS

п/п	Фактор конкурентоспроможності	али 1- 20	Рейтинг товарів-конкурентів у порівнянні з RFS						
			3	2	1		1	2	3
	Інновації	2							
	Функціонал	8							
	Цінова політика	7							

	Ресурсоємність	0							
--	----------------	---	--	--	--	--	--	--	--

SWOT-аналіз проекту наведено в таблиці 4.12.

Таблиця 4.12 - SWOT-аналіз стартап-проекту

Сильні сторони: розумна цінова політика, функціонал забезпечує рішення більшості задач клієнта	Слабкі сторони: відсутність співпраці з інноваційними центрами, велика ресурсоємність
Можливості: впровадження інноваційних рішень, оптимізація роботи продукту	Загрози: неоптимізована робота продукту, узагальнені рішення, що непридатні для вирішення клієнтських задач

Альтернативи ринкового впровадження проекту розглянуто в таблиці 4.13.

Таблиця 4.13 - Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Спеціалізовані рішення	Висока	6-9 місяців
2	Хмарний сервіс	Висока	9-12 місяців
3	Узагальнення рішення, вихід на нові сфери ринку	Середня	12-15 місяці

#### 4.4 Розроблення ринкової стратегії проекту

Опис та вибір цільових груп потенційних клієнтів зображено в таблиці 4.14.

Таблиця 4.14 - Вибір цільових груп потенційних споживачів

п/п	Опис цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
	Малий бізнес	Низька готовність, через необхідність вкладання великих коштів.	Низький попит	Середня	Вхід в сегмент складний
	Середній бізнес	Середня готовність. В залежності від виду бізнесу, готовність різниться.	Середній попит	Вища середньої	Вхід в сегмент достатньо складний
	Великий бізнес	Абсолютна готовність. Більшість компаній такого рівня застосовували чи планують застосовувати інтелектуальні системи прогнозування, тому повністю готові розглянути альтернативні рішення	Високий попит	Середня	Вхід в сегмент складний
Які цільові групи обрано: 2,3					

В таблиці 4.15 зображено вибір базової стратегії розвитку.

Таблиця 4.15 - Визначення базової стратегії розвитку

п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспромо жні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
	Розробка та створення додаткових функціональних модулів	Таргетні пропозиції бізнесу, проведення презентації функціональних рішень на ярмарках та конференціях	Відсутність аналогічних до новостворених функціональних модулів у конкурентів	Розробка та удосконалення існуючих модулів на основі потреб ринку та інформації від клієнтів

В таблиці 4.16 наведено визначення базової стратегії конкурентної поведінки.

Таблиця 4.16 - Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Можливі обидва варіанти	Стандартні функціональні модулі будуть виконувати схожі функції, що і конкуренти	Унікальна цінова політика, функціональні інновації

В таблиці 4.17 наведено визначення стратегії позиціонування.

Таблиця 4.17 - Визначення стратегії позиціонування

п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспро можні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
	Висока якість прогнозування в клієнтській сфері застосування	Розробка та удосконалення існуючих модулів на основі потреб ринку та інформації від клієнтів	Спеціалізо вані рішення, хмарні сервіси	Прогнозуван ня, передбачення, аналіз

#### 4.5 Розроблення маркетингової програми стартап-проекту

В таблиці 4.18 представлені ключові переваги концепції потенційного товару.

Таблиця 4.18 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Широкий функціонал	Вирішення задач	Забезпечує вирішення більшої кількості задач бізнесу
2	Спеціалізовані рішення	Вирішення задач	Забезпечує більш ефективне вирішення задач у звуженій сфері застосування
3	Технічні ресурси	Хмарні сервіси	Дозволяє користуватись рішенням за рахунок віддалених технічних потужностей

Опис трьох рівнів моделі товару відображено у таблиці 4.19.

Таблиця 4.19 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за здумом	Обробка, аналіз даних. Прогнозування та передбачення потреб споживача		
	Властивості/характеристик и	М/ Нм	Вр/Тх /Тл/Е/Ор

II. Товар у реальному виконанні	1. Швидкодія	Нм	Тх/Тл/Е
	2. Ефективність	Нм	Тх/Тл
	3. Користувацький інтерфейс	Нм	Е
	Якість: стандарти відповідні до законодавства. Створені функціональні скріпти.		
	Пакування: Власний сайт		
	Марка: RFS solutions, RFS		

Продовження таблиці 4.19

III. Товар із підкріпленням	До продажу: застосунок для інтеграції в існуючі системи керування підприємством для прогнозування та передбачення потреб споживачів на основі великих масивів даних
	Після продажу: Швидкодія, ефективність, легкість у користуванні
Закритий код. Захищений від можливості декомпіляції.	

Визначення меж встановлення ціни показано в таблиці 4.20.

Таблиця 4.20 - Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	-	200\$/міс	Рівень доходів підприємств надзвичайно високий	150-200\$/міс

Формування системи збуту зображено в таблиці 4.21.

Таблиця 4.21 - Формування системи збуту



№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	Таргетні пропозиції для компаній	Презентації функціоналу	-	-

Концепція маркетингових комунікацій відображена у таблиці 4.22.

Таблиця 4.22 - Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	р
1	Середній бізнес – оптимальні рішення за невисоку ціну	Соціальні мережі, внутрішньо ринкова комунікація	Прогнозування покупок споживача	Короткий опис переваг продукту, заохочення дізнатись більше	покуп
2	Великий бізнес – повноцінні рішення для покращення продажів	Таргетні дзвінки до клієнтів	Прогнозування покупок споживача	Донести інформацію про оптимальність рішення для бізнесу клієнта	покуп

## Висновки до розділу

Отже, відповідно до вищенаведених результатів, можна стверджувати про наявність попиту на запропоновану систему. Варто зауважити, що на ринку присутня сильна конкуренція, але інноваційна складова продукту дозволяє суттєво збільшити конкурентоспроможність проекту. Подальша імплементація проекту доцільна за умови фокусування на середній та великий бізнес, оскільки саме цей ринок динамічно зростає і є достатньо рентабельним.

## ВИСНОВКИ

В даній роботі розглянуто підходи до аналізу даних роздрібної торгівлі, пошуку асоціацій, моделювання та прогнозування основних економічних показників підприємства.

В результаті виконання роботи спроектовано архітектуру СППР для аналізу транзакцій, пошуку залежностей між товарами та їх категоріями, моделювання та прогнозування попиту і прибутків підприємства. Реалізований метод Apriori для пошуку асоціативних правил та методи регресійного аналізу для побудови моделей і прогнозування: AR, ARMA, ARIMA, моделей у вигляді тренду.

Описані методи реалізовані у вигляді програмних модулів зі створеним користувацьким інтерфейсом.

Програмний продукт застосовано на прикладі транзакцій роздрібного магазину. Визначено найбільш вагомі категорії товарів з точки зору прибутковості. Згенеровано асоціативні правила, що показують зв'язки між категоріями товарів. Проведений аналіз та відбір найкращих моделей для прогнозування попиту, прибутку та завантаженості підприємства.

Подальшим напрямком розвитку роботи може бути проведення додаткового аналізу інших існуючих моделей для уточнення прогнозу.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Багиев Г.Л. Маркетинг: учебник для вузов / Багиев Г.Л., Тарасевич В.М. – СПб. : Питер, 2010. – 576 с.
2. Коротков А. В. Управление маркетингом : учебное пособие / Коротков А.В., Синяева И.М. – М. : Юнити-Дана, 2005. – 463 с.
3. Наумов В.Н. Маркетинговые решения в розничной торговле: учебное пособие / Наумов В.Н., Кукура С.П. – СПб. : Политехника-сервис, 2008. – 198 с.
4. The concept of development of the payment system in the Eurasian economic union [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/v/kontseptsiya-razvitiya-platezhnoy-sistemy-v-evraziyskom-ekonomicheskom-soyuze>
5. Дихтль Е. Практический маркетинг: учебное пособие/ Дихтль Е., Хершген Х. – М. : Высш.шк., 2007. – 255 с.
6. Бузукова Е. А. Ассортимент розничного магазина: методы анализа и практические советы / Бузукова Е. А. – СПб. : Питер, 2007. – 176 с.
7. Федько В. П. Основы маркетинга / Федько В. П., Федько Н. Г. – Ростов н/Д : Феникс, 2005. – 497 с.
8. Чкалова О. В. Формирование и развитие розничной торговой сети в мегаполисе : моногр. / Чкалова О. В. – Н. Новгород : ИКИ, 2008. – 231 с.
9. Дашков Л. П. Коммерция и технология торговли: учебник / Дашков Л. П., Памбухчиянц В. К. – М. : Дашков и К°, 2006. – 515 с.
10. Брагин Л. А. Торговое дело: экономика, маркетинг, организация: учебник / Брагина Л. А., Данько Т. П. – М. : Инфра-М, 2006. – 172 с.
11. Панкратов Ф. Г. Коммерческая деятельность: учебник для вузов/ Панкратов Ф. Г., Серегина Т. К. – М. : Дашков и К°, 2006. – 503 с.

12. Снегирева В. В. Розничный магазин: управление ассортиментом по товарным категориям / Снегирева В. В. – СПб. : Питер, 2005. – 416 с.
13. Гуняков Ю. В. Коммерция. Теория коммерческой деятельности/ Гуняков Ю. В. – Красноярск : КГТЭИ, 2008. – 178 с.
14. Modern system of marketing communications and structural correlation of its elements [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/v/instrumenty-i-metody-upravleniya-assortimentom-tovarov-v-riteyle>
15. Применение технологий интеллектуального анализа данных в естественнонаучных, технических и гуманитарных областях [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/v/primenenie-tehnologiy-intellektualnogo-analiza-dannyh-v-estestvennonauchnyh-tehnicheskikh-i-gumanitarnyh-oblastyah>
16. Интеллектуальные информационные технологии проектирования автоматизированных систем диагностирования и распознавания образов: монография / [Субботин С.А., Олейник А.А., Гофман Е.А. и др.] ; под ред. С.А. Субботина. – Х: ООО "Компания Смит", 2012. – 317 с.
17. Айвазян С.А. Прикладная статистика: Исследование зависимостей / С.А. Айвазян, И.С. Енюков, Л.Д. Мешалкин. – М.: Финансы и статистика, 1985. – 487 с.
18. Zhang C. Association rule mining: models and algorithms / C. Zhang, S. Zhang. – Berlin: Springer-Verlag. – 2002. – 238 p.
19. Shin Y.C. Intelligent systems: modeling, optimization, and control / C.Y. Shin, C. Xu. – Boca Raton: CRC Press, 2009. – 456 p.
20. Adamo J.-M. Data mining for association rules and sequential patterns: sequential and parallel algorithms / J.-M. Adamo. – New York: Springer-Verlag. – 2001. – 259 p.
21. Паклин Н.Б. Бизнес-аналитика: от данных к знаниям / Паклин Н.Б., Орешков В.И. – СПб.: Питер. - 2013. - 704 с.

22. Бідюк П. І. Часові ряди: моделювання і прогнозування / Бідюк П.І., Савенков О. І., Баклан І.В. – К.: ЕКМО, 2003. – 144 с.
23. Бокс Дж., Анализ временных рядов. Прогноз и управление / Дж. Бокс, Г. Дженкинс - М.: Мир, 1974. - 402 с.
24. Бідюк П.І. Принципи прогнозування часових рядів / Бідюк П.І., Шехтер Д.В., Клименко О.М. // Наукові вісті НТУУ „КПІ”. – 2005. – № 5. – С. 14-25.
25. Бидюк П.И. Курс лекций по анализу временных рядов [Текст]. / П.И. Бидюк– К.: НТУУ «КПИ», 2009. – 450 с.
26. Ставицький А.В. Навчально-методичний комплекс з курсів „Прогнозування” та „Фінансове прогнозування” / А.В. Ставицький. – К.: РВВ ІМФ, 2006. – 107 с.
27. Бідюк П. І. Аналіз часових рядів (навчальний посібник) / П. І. Бідюк, В. Д. Романенко, О. Л.Тимошук. – К.: Політехніка, 2010. – 317с.

## ДОДАТОК А ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДОПОВІДІ



## Побудова системи підтримки прийняття рішень для підприємств роздрібної торгівлі на основі методів інтелектуального аналізу даних

Науковий керівник: Данилов Валерій Якович

Доповідач: Тарасюк Тарас Сергійович



1



## Актуальність проблеми

У зв'язку зі зниженням питомої вартості зберігання даних, зростає обсяг інформації, що накопичують підприємства роздрібної торгівлі. Виникають завдання, пов'язані з необхідністю обробки великих масивів даних з метою пошуку нових латентних закономірностей, встановлення і виявлення нових знань.

В магістерській дисертації здійснюється проектування та реалізація системи підтримки прийняття рішень, що забезпечує пошук асоціацій, моделювання та прогнозування ключових фінансових показників підприємства та сприяє створенню стратегій розвитку підприємства.

2

## Об'єкт, предмет та мета дослідження

- Мета дослідження – розробка структури СППР та її реалізація у вигляді програмного продукту для побудови асоціацій, моделювання та прогнозування бізнес-процесів у роздрібній торгівлі.
- Об'єкт дослідження – статистичні ряди даних бізнес-процесів роздрібної торгівлі.
- Предмет дослідження – моделі та методи інтелектуального аналізу даних: алгоритм Apriori для генерації асоціативних правил, моделі регресійного аналізу з прогнозування часових рядів: AR, ARMA, ARIMA та моделей у вигляді тренду.

3

## Задачі дослідження

- Аналіз потреб підприємств роздрібної торгівлі
- Дослідження методів інтелектуального аналізу даних в роздрібній торгівлі
- Побудова системи підтримки прийняття рішень
- Застосування системи підтримки прийняття рішення
- Аналіз отриманих результатів

4



## Маркетингові рішення в роздрібній торгівлі

- вибір цільового ринку;
- формування товарного асортименту і комплексу послуг;
- Побудова дорожньої карти магазину;
- вибір методів ціноутворення і стимулювання збуту;
- вибір розміщення магазину.

5

## Задачі СППР у роздрібній торгівлі

- Аналіз і порівняння прибутку
- Порівняння історії продажів
- Аналіз ефективності просування продукції
- Націлювання клієнтів
- Керування запасами
- Аналіз кошика споживача
- Створення моделей прогнозування

6

## Методи ІАД в роздрібній торгівлі

- Асоціативний аналіз
- Аналіз та прогнозування часових рядів

7

## Асоціативний аналіз в роздрібній торгівлі

Побудова асоціативних правил дозволяє ефективно вирішувати наступні задачі:

- визначення товарів, які варто просувати спільно;
- вибір місця розташування товару в магазині;
- аналіз споживчого кошика;

8

## Аналіз та прогнозування часових рядів

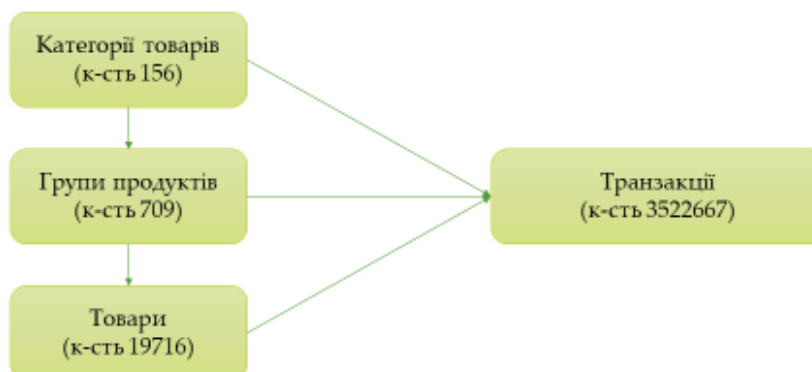
Інструментарій прогнозування часових рядів у роздрібній торгівлі дозволяє вирішити наступні задачі:

- прогнозування попиту на товари або їх категорій;
- прогнозування завантаженості магазину;
- прогнозування прибутків.

9

## Застосування СППР на прикладі магазину роздрібної торгівлі

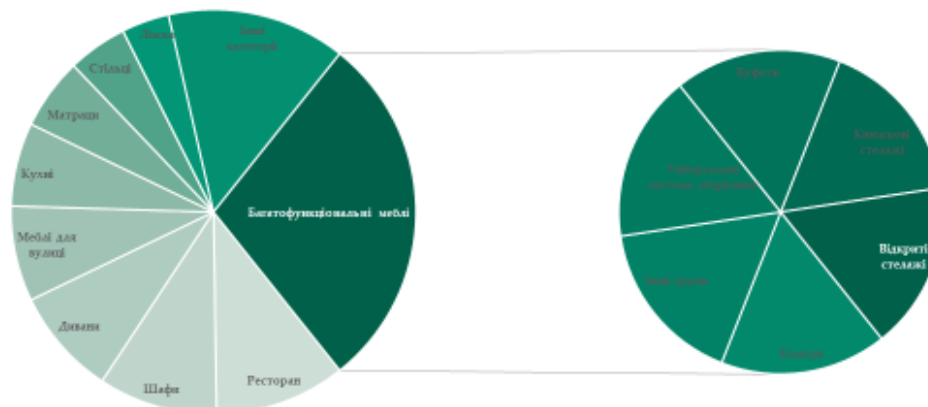
Ієрархія даних магазину:



10

## Структура даних магазину

Категорії товарів -> групи продуктів



11

## Структура даних магазину

Детальна інформація про транзакцію



12

## Опис вихідних даних

Code	Item Category	Code	Product Group	No.	Item
11	Sofas	111	Sofas	81591	MONGSTAD MIRROR 94X190 BLACK-BROWN
12	Armchairs, footstool & sofa ta	112	Leather sofas		FAVORIT SAUCEPAN/LID 2 L STAINLESS STEEL
14	Occasional tables range	113	Sofa beds.	83463	
15	Solitaire sofa-beds, armchairs	114	Armchairs	84325	LINDMON VEN BLIND 60X155 BROWN
16	Seating & Reclining Furniture	115	Footstools.	85711	BILLY BOOKCS 40X28X202 BLACK-BROWN
21	Multipurpose furniture	119	Furniture fabrics	88979	SANTIR N WLL UPLT GLASS/STEEL
22	Store and display furniture	121	Sofa tables	89163	KONCIS N GARLIC PRESS STAINLESS STEEL
23	Tv furniture & electronics	123	Glass-door cabinets, shelving	91415	TROFAST LID 20X28 WHITE
24	Living-room Storage	124	Audio & video furniture	94089	BILLY N BOOKCS 40X28X202 MEDIUM BROWN
...	...	...	...	...	...

13

## Опис вихідних даних

Transaction No.	Item No.	Item Category Code	Product Group Code	Price	Net Price	Quantity	Date	Customer No.
1010007738	11536	76	762	2,29	1,89	1	6-1-16	CUST02000320
1010007738	133038	141	1414	5,99	4,95	1	6-1-16	CUST02000320
1010007738	141401	94	942	0,99	0,82	1	6-1-16	CUST02000320
1010007738	149459	142	1421	4,99	4,12	1	6-1-16	CUST02000320
1010007738	150004	111	1111	4,49	3,71	1	6-1-16	CUST02000320
1010007738	152159	142	1422	3,99	3,3	1	6-1-16	CUST02000320
1010007738	152282	82	822	29,99	24,79	1	6-1-16	CUST02000320
1010007738	165213	93	936	29,99	24,79	1	6-1-16	CUST02000320
1010007738	177225	76	761	2,29	1,89	1	6-1-16	CUST02000320

14



## Асоціативні правила

Rules	Support	Confidence	Improvement
{ 111 } => { 164 }	0,0001	0,4402	3967,541
{ 154 } => { 164 }	0,0001	0,179	1613,261
{ 091 } => { 164 }	0,0001	0,0496	447,4992
{ 121 } => { 164 }	0,0001	0,005	44,8734
{ 154 } => { 111 }	0,0003	0,4066	1613,261
{ 091 } => { 111 }	0,0003	0,1128	447,4992
{ 121 } => { 111 }	0,0003	0,0113	44,8734
{ 091 } => { 154 }	0,0006	0,2774	447,4992
{ 121 } => { 154 }	0,0006	0,0278	44,8734
{ 121 } => { 091 }	0,0022	0,1003	44,8734

Rules	Support	Confidence	Improvement
{ 111 } => { 164, 154 }	0,0001	0,4402	3967,541
{ 154 } => { 164, 111 }	0,0001	0,179	1613,261
{ 111 } => { 164, 091 }	0,0001	0,4402	3967,541
{ 111, 091 } => { 164 }	0,0001	0,4402	3967,541
{ 091 } => { 164, 111 }	0,0001	0,0496	447,4992
{ 111 } => { 164, 121 }	0,0001	0,4402	3967,541
{ 111, 121 } => { 164 }	0,0001	0,4402	3967,541
{ 091 } => { 154, 121 }	0,0006	0,2774	447,4992
{ 091, 121 } => { 154 }	0,0006	0,2774	447,4992
{ 121 } => { 154, 091 }	0,0006	0,0278	44,8734

17

## Регресійні моделі

AP(5):

$$Y = 8149,4416 + 0,1695 * y(k-2) + 0,1128 * y(k-3) + 0,1078 * y(k-4) + 0,4334 * y(k-5)$$

RMSE	Коефіцієнт детермінації	ІКА	Байєса-Шварца	Фішера	Дарбіна-Уотсона
2844.81	0.223	137.25	135.3	0.303	2.308

APKC(5,4)

$$Y = 8149,4416 + 0,1695 * y(k-2) + 0,1128 * y(k-3) + 0,1078 * y(k-4) + 0,4334 * y(k-5) + 1882,9583 - 3,1462 * ma(k-1) - 0,5581 * ma(k-2) - 0,0319 * ma(k-3) - 0,4448 * ma(k-4)$$

RMSE	Коефіцієнт детермінації	ІКА	Байєса-Шварца	Фішера	Дарбіна-Уотсона
2253.12	0.582	94.92	92.97	1.394	2.79

18

## Регресійні моделі

АРІКС(5,4,3):

$$Y = 576,4805 - 0,488 * y(k-2) - 0,492 * y(k-3) - 0,49 * y(k-4) - 0,4693 * y(k-5) + 11428,0654 - 2,5581 * ma(k-1) - 0,0425 * ma(k-2) + 0,469 * ma(k-3) + 0,0677 * ma(k-4) - 1,29 * ma(k-5) + y(k) - 3 * y(k-1) + 3 * y(k-2) - y(k-3)$$

RMSE	Коефіцієнт детермінації	ІКА	Байєса-Шварца	Фішера	Дарбіна-Уотсона
2860.6	0.582	97.547	95.59	1.394	1.731

Тренд порядку 3

$$Y(k) = 38108,1527 - 247,4516 * k + 40,7523 * k^2 - 0,8355 * k^3$$

RMSE	Коефіцієнт детермінації	ІКА	Байєса-Шварца	Фішера	Дарбіна-Уотсона
2556.71	0.336	174.776	172.823	0.506	2.653

19

## Висновки

- Досліджено методи інтелектуального аналізу даних
- Побудована система підтримки прийняття рішень, що базується на методах інтелектуального аналізу даних
- Застосовано систему підтримки прийняття рішення для роздрібного магазину. А саме, згенеровані асоціативні правила, що показують залежності між товарами та побудовані моделі для прогнозування попиту на категорії товарів, прибутків та завантаженості магазину.

20



## ДОДАТОК Б ЛІСТИНГ КОДУ

```

using RetailDSS.Models;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Windows;
using RetailDSS.Models.Business_Logic;
using RetailDSS.Business_Logic;
using Microsoft.Win32;
using System.Windows.Input;

namespace RetailDSS
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        List<Transaction> transactions;
        List<Item_Category> itemCategories;
        List<Product_Group> productGroups;
        List<Item> items;
        Tuple<List<Item_Category>, List<Item_Category>, List<Item_Category>, List<string>> ABCItemCategories;
        Tuple<List<Product_Group>, List<Product_Group>, List<Product_Group>, List<string>> ABCProductGroups;
        Tuple<List<Item>, List<Item>, List<Item>, List<string>> ABCItems;

        public MainWindow()
        {
            InitializeComponent();
        }

        private void button_Click(object sender, RoutedEventArgs e)
        {
            //List<Transaction> transactions = Calculations.GetTransactions(@"C:\Users\ttarsiuk\Documents\SY\Disertation\Data\Test
Transactions.csv");

            //List<Item_Category> itemCategoriesTest = new List<Item_Category>()
            //{
            //    new Item_Category("31", "Test1"),
            //    new Item_Category("951", "Test2"),
            //    new Item_Category("953", "Test3")
            //};
            // ABC
            //List<string> linesWithProfit = Calculations.ABCItemCategories(transactions, itemCategories, true);
            //File.WriteAllLines(@"C:\Users\ttarsiuk\Documents\SY\Disertation\Data\Results\ABC      Category      Profit.csv",
linesWithProfit);
            //Tuple<List<Item_Category>, List<Item_Category>, List<Item_Category>, List<string>> ABCItemCategoriesQty =
Calculations.ABCItemCategories(transactions, itemCategories, false);
            //List<Item_Category> itemCategoriesA = ABCItemCategoriesQty.Item1;
            //File.WriteAllLines(@"C:\Users\ttarsiuk\Documents\SY\Disertation\Data\Results\ABC      Category      Quantities.csv",
linesWithQuantities.Item2);

            //List<string> linesWithProfit = Calculations.ABCProductGroups(transactions, productGroups, true);
            //File.WriteAllLines(@"C:\Users\ttarsiuk\Documents\SY\Disertation\Data\Results\ABC      Product      Group      Profit.csv",
linesWithProfit);
            //List<string> linesWithQuantities = Calculations.ABCProductGroups(transactions, productGroups, false);
            //File.WriteAllLines(@"C:\Users\ttarsiuk\Documents\SY\Disertation\Data\Results\ABC      Product      Group      Quantities.csv",
linesWithQuantities);

            //List<string> linesWithProfit = Calculations.ABCItems(transactions, items, true);
            //File.WriteAllLines(@"C:\Users\ttarsiuk\Documents\SY\Disertation\Data\Results\ABC Item Profit.csv", linesWithProfit);
            //List<string> linesWithQuantities = Calculations.ABCItems(transactions, items, false);
            //File.WriteAllLines(@"C:\Users\ttarsiuk\Documents\SY\Disertation\Data\Results\ABC      Item      Quantities.csv",
linesWithQuantities);

            // Split
            //Calculations.SplitTransactionsByMonth(transactions);

```

```

//Calculations.SplitTransactionsByWeek(transactions);

// Regression Data
//Calculations.GenerateRegressionData(transactions, itemCategories, false);
//Calculations.GenerateRegressionData(transactions, itemCategories, true);

// Apriori
//Apriori.RunForItemCategories(transactions, itemCategoriesA, 0.0001);
//List<Tuple<Item_Category, List<Item>>> itemsByCategories = Calculations.GetItemsByCategories(itemCategoriesA,
items);

//var transactonsByCategories = Calculations.GetTransactionsByCategories(itemCategoriesA, transactions);
//foreach (var item in itemsByCategories)
//{
//    var transactionsByCategory = transactonsByCategories.Select(x => x).Where(x => x.Item1.Equals(item.Item1)).ToList();
//    Apriori.RunForItems(transactionsByCategory[0].Item2, item.Item2, 0.05, item.Item1);
//}

// Regression
//List<double> Y = new List<double>()
//{
//    40079.25,
//    36311.5,
//    36349.25,
//    37324,
//    36778.25,
//    38641,
//    37149,
//    38543.84,
//    41749.25,
//    39956,
//    45877.3,
//    31769,
//    40141,
//    42161,
//    41027.5,
//    42818.6,
//    40435.5,
//    40896,
//    43526.3,
//    43951.75,
//    39382,
//    41765,
//    46864.1,
//    42579,
//    37466,
//    43119.5
//};
//for (int i = 2; i < 6; i++)
//{
//    Regression.ARMA(Y.Count, Y, i, 2, 0.1, 0.1, Regression.MovAv.Previous);
//}
//Regression.AR(Y.Count, Y, 8, 0.1);
//Regression.Trend(Y.Count, Y, 2);

//Calculations.GetRegressionDataByCategories(transactions, itemCategoriesA);
//List<Tuple<Item_Category, List<Item>>> itemsByCategories = Calculations.GetItemsByCategories(itemCategoriesA,
items);
//List<Tuple<Item, List<double>, List<double>>> regressionDataByItems =
Calculations.GetRegressionDataByItems(transactions, itemsByCategories[0].Item2);
//Regression.AR(regressionDataByItems[0].Item2.Count, regressionDataByItems[0].Item2, 6, 0.1);
//List<Tuple<Item_Category, List<double>, List<double>>> regressionDataByCategories =
Calculations.GetRegressionDataByCategories(transactions, itemCategoriesA);
//Regression.AR(regressionDataByCategories[0].Item2.Count, regressionDataByCategories[0].Item2, 5, 0.1);
}

private void MenuItem1_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Multiselect = true;
    openFileDialog.Filter = "CSV files (*.csv)|*.csv|All files (*.*)|*.*";
    openFileDialog.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
    if (openFileDialog.ShowDialog() == true)
    {
        using (new WaitCursor())
    }
}

```

```

        {
            transactions = Calculations.GetTransactions(openFileDialog.FileName);

            TransactionsGrid.ItemsSource = transactions;
        }
        MessageBox.Show("Транзакції успішно завантажені!", "RSS");
    }
}

private void MenuItem2_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Multiselect = true;
    openFileDialog.Filter = "CSV files (*.csv)|*.csv|All files (*.*)|*.*";
    openFileDialog.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
    if (openFileDialog.ShowDialog() == true)
    {
        using (new WaitCursor())
        {
            itemCategories = Calculations.GetItemCategories(openFileDialog.FileName);
            ItemCategoriesGrid.ItemsSource = itemCategories;
        }
        MessageBox.Show("Категорії товарів успішно завантажені!", "RSS");
    }
}

private void MenuItem3_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Multiselect = true;
    openFileDialog.Filter = "CSV files (*.csv)|*.csv|All files (*.*)|*.*";
    openFileDialog.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
    if (openFileDialog.ShowDialog() == true)
    {
        using (new WaitCursor())
        {
            productGroups = Calculations.GetProductGroups(openFileDialog.FileName);
            ProductGroupsGrid.ItemsSource = productGroups;
        }
        MessageBox.Show("Групи продуктів успішно завантажені!", "RSS");
    }
}

private void MenuItem4_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Multiselect = true;
    openFileDialog.Filter = "CSV files (*.csv)|*.csv|All files (*.*)|*.*";
    openFileDialog.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
    if (openFileDialog.ShowDialog() == true)
    {
        using (new WaitCursor())
        {
            items = Calculations.GetItems(openFileDialog.FileName);
            ItemsGrid.ItemsSource = items;
        }
        MessageBox.Show("Товари успішно завантажені!", "RSS");
    }
}

private void button1_Click(object sender, RoutedEventArgs e)
{
    if (Convert.ToDouble(textBox1.Text) + Convert.ToDouble(textBox2.Text) + Convert.ToDouble(textBox3.Text) == 1)
    {
        switch (comboBox1.SelectedIndex)
        {
            case 0:
                if (comboBox2.SelectedIndex == 0)
                {
                    ABCItemCategories = Calculations.ABCItemCategories(transactions, itemCategories, false,
Convert.ToDouble(textBox1.Text), Convert.ToDouble(textBox2.Text), Convert.ToDouble(textBox3.Text));
                }
                else
                {
                    ABCItemCategories = Calculations.ABCItemCategories(transactions, itemCategories, true,
Convert.ToDouble(textBox1.Text), Convert.ToDouble(textBox2.Text), Convert.ToDouble(textBox3.Text));
                    AElementsGrid.ItemsSource = ABCItemCategories.Item1;
                    BElementsGrid.ItemsSource = ABCItemCategories.Item2;
                    CElementsGrid.ItemsSource = ABCItemCategories.Item3;
                    break;
                }
            }
        }
    }
}

```

```

        case 1:
            if (comboBox2.SelectedIndex == 0)
                ABCProductGroups = Calculations.ABCProductGroups(transactions, productGroups, false,
Convert.ToDouble(textBox1.Text), Convert.ToDouble(textBox2.Text), Convert.ToDouble(textBox3.Text));
            else
                ABCProductGroups = Calculations.ABCProductGroups(transactions, productGroups, true,
Convert.ToDouble(textBox1.Text), Convert.ToDouble(textBox2.Text), Convert.ToDouble(textBox3.Text));
                AElementsGrid.ItemsSource = ABCProductGroups.Item1;
                BElementsGrid.ItemsSource = ABCProductGroups.Item2;
                CElementsGrid.ItemsSource = ABCProductGroups.Item3;
                break;
        case 2:
            if (comboBox2.SelectedIndex == 0)
                ABCItems = Calculations.ABCItems(transactions, items, false, Convert.ToDouble(textBox1.Text),
Convert.ToDouble(textBox2.Text), Convert.ToDouble(textBox3.Text));
            else
                ABCItems = Calculations.ABCItems(transactions, items, true, Convert.ToDouble(textBox1.Text),
Convert.ToDouble(textBox2.Text), Convert.ToDouble(textBox3.Text));
                AElementsGrid.ItemsSource = ABCItems.Item1;
                BElementsGrid.ItemsSource = ABCItems.Item2;
                CElementsGrid.ItemsSource = ABCItems.Item3;
                break;
        default:
            break;
    }
    else
        MessageBox.Show("Сума ваг категорій повинна бути рівна 1!", "Помилка!");
}

private void button2_Click(object sender, RoutedEventArgs e)
{
    switch (comboBox3.SelectedIndex)
    {
        case 0:
            var itemCategoriesForRules = new List<Item_Category>();
            if (ABCItemCategories != null)
                itemCategoriesForRules = ABCItemCategories.Item1;
            else
                itemCategoriesForRules = itemCategories;
            var categoryRules = Apriori.GetRulesForCategories(transactions, itemCategoriesForRules,
Convert.ToDouble(textBox4.Text), Convert.ToDouble(textBox5.Text));
            ElementsGrid1.ItemsSource = categoryRules.Item1;
            ElementsGrid2.ItemsSource = categoryRules.Item2;
            ElementsGrid3.ItemsSource = categoryRules.Item3;
            break;
        case 1:
            AElementsGrid.ItemsSource = ABCProductGroups.Item1;
            break;
        case 2:
            string categoryCode = textBox.Text;
            var itemRules = Apriori.GetRulesForItems(transactions, items.Select(x => x).Where(x => x.ItemCategoryCode ==
categoryCode).ToList(), Convert.ToDouble(textBox4.Text), Convert.ToDouble(textBox5.Text), new Item_Category(categoryCode, ""));
            ElementsGrid1.ItemsSource = itemRules.Item1;
            ElementsGrid2.ItemsSource = itemRules.Item2;
            ElementsGrid3.ItemsSource = itemRules.Item3;
            break;
        default:
            break;
    }
}

public class WaitCursor : IDisposable
{
    private Cursor _previousCursor;

    public WaitCursor()
    {
        _previousCursor = Mouse.OverrideCursor;

        Mouse.OverrideCursor = Cursors.Wait;
    }

    #region IDisposable Members

    public void Dispose()

```

```

    {
        Mouse.OverrideCursor = _previousCursor;
    }

#endregion
}
}

<Window x:Class="RetailDSS.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:RetailDSS"
    xmlns:vm="clr-namespace:RetailDSS.ViewModels"
    xmlns:wpfTool="clr-namespace:Xceed.Wpf.Toolkit;assembly=Xceed.Wpf.Toolkit"
    xmlns:lvc="clr-namespace:LiveCharts.Wpf;assembly=LiveCharts.Wpf"
    mc:Ignorable="d"
    Title="Retail Decision Support System" Height="600" Width="1350">

<Window.Resources>
    <vm:TransactionViewModel x:Key="TransactionViewModel"/>
</Window.Resources>
<Grid>
    <DockPanel>
        <Menu DockPanel.Dock="Top">
            <MenuItem Header="Завантажити дані">
                <MenuItem Header="Транзакції" Click="MenuItem1_Click" />
                <MenuItem Header="Категорії товарів" Click="MenuItem2_Click" />
                <MenuItem Header="Групи продуктів" Click="MenuItem3_Click" />
                <MenuItem Header="Товари" Click="MenuItem4_Click" />
            </MenuItem>
            <MenuItem Header="Про програму" />
        </Menu>
        <TabControl Margin="10,10,10,10">
            <TabItem Header="Дані">
                <Grid>
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition Width="2*" />
                        <ColumnDefinition Width="1*" />
                        <ColumnDefinition Width="1*" />
                        <ColumnDefinition Width="1*" />
                    </Grid.ColumnDefinitions>
                    <Grid.RowDefinitions>
                        <RowDefinition />
                        <RowDefinition Height="18*" />
                    </Grid.RowDefinitions>
                    <Label HorizontalAlignment="Center" Content="Транзакції" Grid.Column="0" Grid.Row="0"/>
                    <Label HorizontalAlignment="Center" Content="Категорії товарів" Grid.Column="1" Grid.Row="0"/>
                    <Label HorizontalAlignment="Center" Content="Групи продуктів" Grid.Column="2" Grid.Row="0"/>
                    <Label HorizontalAlignment="Center" Content="Товари" Grid.Column="3" Grid.Row="0"/>
                    <DataGrid Name="TransactionsGrid" Margin="5,5,5,5" Grid.Column="0" Grid.Row="1"/>
                    <DataGrid Name="ItemCategoriesGrid" Margin="5,5,5,5" Grid.Column="1" Grid.Row="1"/>
                    <DataGrid Name="ProductGroupsGrid" Margin="5,5,5,5" Grid.Column="2" Grid.Row="1"/>
                    <DataGrid Name="ItemsGrid" Margin="5,5,5,5" Grid.Column="3" Grid.Row="1"/>
                </Grid>
            </TabItem>
            <TabItem Header="ABC-аналіз">
                <Grid>
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition Width="1*" />
                        <ColumnDefinition Width="1*" />
                        <ColumnDefinition Width="1*" />
                    </Grid.ColumnDefinitions>
                    <Grid.RowDefinitions>
                        <RowDefinition Height="1.25*" />
                        <RowDefinition />
                        <RowDefinition Height="18*" />
                    </Grid.RowDefinitions>
                    <Label HorizontalAlignment="Center" Content="Категорія - А" Grid.Column="0" Grid.Row="1"/>
                    <Label HorizontalAlignment="Center" Content="Категорія - В" Grid.Column="1" Grid.Row="1"/>
                    <Label HorizontalAlignment="Center" Content="Категорія - С" Grid.Column="2" Grid.Row="1"/>
                    <DataGrid Name="AElementsGrid" Margin="5,5,5,5" Grid.Column="0" Grid.Row="2" Focusable="False"/>
                    <DataGrid Name="BElementsGrid" Margin="5,5,5,5" Grid.Column="1" Grid.Row="2" Focusable="False"/>
                    <DataGrid Name="CElementsGrid" Margin="5,5,5,5" Grid.Column="2" Grid.Row="2" Focusable="False"/>
                </Grid>
            </TabItem>
        </TabControl>
    </DockPanel>
</Grid>

```

```

        <ComboBox x:Name="comboBox1" SelectedIndex="0" HorizontalAlignment="Left" Margin="5,5,0,0"
VerticalAlignment="Top" Width="125" Height="21">
        <ComboBoxItem>Категорії товарів</ComboBoxItem>
        <ComboBoxItem>Групи продуктів</ComboBoxItem>
        <ComboBoxItem>Товари</ComboBoxItem>
    </ComboBox>
    <ComboBox x:Name="comboBox2" SelectedIndex="0" HorizontalAlignment="Left" Margin="135,5,0,0"
VerticalAlignment="Top" Width="100" Height="21">
    <ComboBoxItem Name="cbi1">По кількості</ComboBoxItem>
    <ComboBoxItem Name="cbi2">По прибутку</ComboBoxItem>
    </ComboBox>
    <Button x:Name="button1" Content="Визначити категорії" HorizontalAlignment="Left" Margin="5,10,0,0"
VerticalAlignment="Top" Width="130" Height="21" Click="button1_Click" Grid.Column="2"/>
    <TextBox x:Name="textBox1" Grid.Column="1" HorizontalAlignment="Left" Height="17" Margin="57,10,0,0"
TextAlignment="Right" TextWrapping="Wrap" Text="0,8" VerticalAlignment="Top" Width="35"/>
    <Label Content="Bara A:" Grid.Column="1" HorizontalAlignment="Left" Margin="10,5,0,0"
VerticalAlignment="Top"/>
    <TextBox x:Name="textBox2" Grid.Column="1" HorizontalAlignment="Left" Height="17" Margin="159,10,0,0"
TextAlignment="Right" TextWrapping="Wrap" Text="0,15" VerticalAlignment="Top" Width="35"/>
    <Label Content="Bara B:" Grid.Column="1" HorizontalAlignment="Left" Margin="107,5,0,0"
VerticalAlignment="Top"/>
    <TextBox x:Name="textBox3" Grid.Column="1" HorizontalAlignment="Left" Height="17" Margin="263,10,0,0"
TextAlignment="Right" TextWrapping="Wrap" Text="0,05" VerticalAlignment="Top" Width="35"/>
    <Label Content="Bara C:" Grid.Column="1" HorizontalAlignment="Left" Margin="211,5,0,0"
VerticalAlignment="Top"/>
    </Grid>
</TabItem>
<TabItem Header="Асоціативний аналіз">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="1.75*"/>
            <ColumnDefinition Width="2*"/>
            <ColumnDefinition Width="2.25*"/>
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="1.25*"/>
            <RowDefinition/>
            <RowDefinition Height="18*" />
        </Grid.RowDefinitions>
        <Label HorizontalAlignment="Center" Content="Асоціативні правила" Grid.Column="0" Grid.Row="1" />
        <Label HorizontalAlignment="Center" Content="Подвійні асоціативні правила" Grid.Column="1" Grid.Row="1" />
        <Label HorizontalAlignment="Center" Content="Потрійні асоціативні правила" Grid.Column="2" Grid.Row="1" />
        <DataGrid Name="ElementsGrid1" Margin="5,5,5,5" Grid.Column="0" Grid.Row="2" Focusable="False"/>
        <DataGrid Name="ElementsGrid2" Margin="5,5,5,5" Grid.Column="1" Grid.Row="2" Focusable="False"/>
        <DataGrid Name="ElementsGrid3" Margin="5,5,5,5" Grid.Column="2" Grid.Row="2" Focusable="False"/>
        <ComboBox x:Name="comboBox3" SelectedIndex="0" HorizontalAlignment="Left" Margin="5,5,0,0"
VerticalAlignment="Top" Width="125" Height="21">
        <ComboBoxItem>Категорії товарів</ComboBoxItem>
        <ComboBoxItem>Групи продуктів</ComboBoxItem>
        <ComboBoxItem>Товари</ComboBoxItem>
    </ComboBox>
    <Button x:Name="button2" Content="Згенерувати асоціативні правила" HorizontalAlignment="Left"
Margin="5,11,0,0" VerticalAlignment="Top" Width="200" Height="21" Click="button2_Click" Grid.Column="2" Grid.RowSpan="2"/>
    <Label Content="Підтримка:" Grid.Column="1" HorizontalAlignment="Left" Margin="10,5,0,0"
VerticalAlignment="Top"/>
    <TextBox x:Name="textBox4" Grid.Column="1" HorizontalAlignment="Left" TextAlignment="Right" Height="16"
Margin="85,10,0,0" TextWrapping="Wrap" Text="0,02" VerticalAlignment="Top" Width="56"/>
    <Label Content="Впевненість:" Grid.Column="1" HorizontalAlignment="Left" Margin="146,5,0,0"
VerticalAlignment="Top" />
    <TextBox x:Name="textBox5" Grid.Column="1" HorizontalAlignment="Left" TextAlignment="Right" Height="16"
Margin="230,10,0,0" TextWrapping="Wrap" Text="0,6" VerticalAlignment="Top" Width="52"/>
    <TextBox x:Name="textBox" HorizontalAlignment="Left" Height="16" Margin="146,9,0,0" TextWrapping="Wrap"
Text="" VerticalAlignment="Top" Width="51"/>
    </Grid>
</TabItem>
<TabItem Header="Регресійний аналіз">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="1.25*"/>
            <ColumnDefinition Width="1.25*"/>
            <ColumnDefinition Width="10*"/>
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition/>
            <RowDefinition Height="18*" />
        </Grid.RowDefinitions>

```

```

<Grid Grid.Row="1" Grid.Column="2" Name="Chart" Panel.ZIndex="1">
  <Ivc:CartesianChart Series="{Binding SeriesCollection4}" LegendLocation="Right" Background="White">
    <Ivc:CartesianChart.AxisY>
      <Ivc:Axis LabelFormatter="{Binding YFormatter}"></Ivc:Axis>
    </Ivc:CartesianChart.AxisY>
    <Ivc:CartesianChart.AxisX>
      <Ivc:Axis Labels="{Binding Labels}"></Ivc:Axis>
    </Ivc:CartesianChart.AxisX>
  </Ivc:CartesianChart>
</Grid>
<Label x:Name="label" Content="Вибірка:" Height="44" Width="60" Margin="10,0,0,0"
HorizontalAlignment="Left" Grid.Column="1" Grid.Row="0"/>
<DataGrid Name="RegressionGrid" Margin="5,5,5,5" Grid.Column="1" Grid.Row="2" Focusable="False"/>
<Button x:Name="button3" Content="Завантажити" HorizontalAlignment="Left" Height="20" Margin="5,5,0,0"
VerticalAlignment="Top" Width="109" Grid.Row="0" Click="button3_Click"/>
<Label x:Name="label1" Content="Об'єм вибірки:" HorizontalAlignment="Left" Height="24" Margin="5,5,0,0"
Grid.Row="1" VerticalAlignment="Top" Width="100"/>
<TextBox x:Name="textBox6" IsEnabled="False" HorizontalAlignment="Left" Height="18" Margin="10,29,0,0"
Grid.Row="1" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="89"/>
<Label x:Name="label2" Content="Навчальна:" HorizontalAlignment="Left" Height="24" Margin="5,45,0,0"
Grid.Row="1" VerticalAlignment="Top" Width="100"/>
<TextBox x:Name="textBox7" HorizontalAlignment="Left" Height="18" Margin="10,70,0,0" Grid.Row="1"
TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="89"/>
<Label x:Name="label3" Content="Тестова:" HorizontalAlignment="Left" Height="24" Margin="5,85,0,0"
Grid.Row="1" VerticalAlignment="Top" Width="100"/>
<TextBox x:Name="textBox8" HorizontalAlignment="Left" Height="18" Margin="10,110,0,0" Grid.Row="1"
TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="89"/>
<Label x:Name="label4" Content="Прогноз:" HorizontalAlignment="Left" Height="24" Margin="5,125,0,0"
Grid.Row="1" VerticalAlignment="Top" Width="100"/>
<TextBox x:Name="textBox9" HorizontalAlignment="Left" Height="18" Margin="10,150,0,0" Grid.Row="1"
TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="89"/>
<Button x:Name="button4" Content="Показати графік" HorizontalAlignment="Left" Height="40"
Margin="5,193,0,0" Grid.Row="1" VerticalAlignment="Top" Width="109" Click="button4_Click"/>
<TabControl x:Name="tabControl" Grid.Column="2" Grid.Row="1" Margin="67,10,10,368">
  <TabItem Header="AP">
    <Grid>
      <Label Content="Порядок моделі:" Height="30" Width="110" Margin="20,22,850,29"></Label>
      <TextBox Margin="130,26,775,26"/>
      <Button Height="30" Width="120" Content="Побудувати модель" Margin="274,20,576,21"
Click="Button_Click"></Button>
    </Grid>
  </TabItem>
  <TabItem Header="APKC">
    </TabItem>
  <TabItem Header="Тренд">
    </TabItem>
  <TabItem Header="APIKC">
    </TabItem>
</TabControl>
<Label Content="Модель:" Grid.Column="2" Height="27" Margin="67,122,931,328" Grid.Row="1"/>
<TextBox Name="Model" Grid.Column="2" Margin="140,125,10,300" Grid.Row="1" TextWrapping="Wrap"
AcceptsReturn="True"></TextBox>
<TabControl Grid.Column="2" Grid.Row="1" Margin="67,193,10,10">
  <TabItem Header="Навчальна">
    <Grid>
      <Grid.RowDefinitions>
        <RowDefinition Height="2*"></RowDefinition>
        <RowDefinition></RowDefinition>
      </Grid.RowDefinitions>
      <Ivc:CartesianChart Series="{Binding SeriesCollection1}" LegendLocation="Right" >
        <Ivc:CartesianChart.AxisY>
          <Ivc:Axis LabelFormatter="{Binding YFormatter}"></Ivc:Axis>
        </Ivc:CartesianChart.AxisY>
        <Ivc:CartesianChart.AxisX>
          <Ivc:Axis Labels="{Binding Labels}"></Ivc:Axis>
        </Ivc:CartesianChart.AxisX>
      </Ivc:CartesianChart>
      <GroupBox Header="Критерії якості моделі для навчальної вибірки" Height="88" Width="600"
Margin="0,0,0,0" Grid.Row="1">
        <Grid>
          <Label Height="25" Width="44" Margin="0,10,525,31" Content="RMSE:"
HorizontalAlignment="Right"></Label>
          <Label Height="25" Width="75" Margin="0,10,450,31" Name="RMSE1"
HorizontalAlignment="Right"></Label>
          <Label Height="25" Width="32" Margin="0,40,525,1" Content="IKA:"
HorizontalAlignment="Right"></Label>

```

```

        <Label Height="25" Width="75" Margin="0,40,450,1" Name="IKA1"
HorizontalAlignment="Right"></Label>
        <Label Height="25" Width="55" Margin="0,10,300,31" Content="Фішера:"
HorizontalAlignment="Right"></Label>
        <Label Height="25" Width="87" Margin="0,10,213,31" Name="F1"
HorizontalAlignment="Right"></Label>
        <Label Height="25" Width="145" Margin="0,40,300,1" Content="Коефіцієнт детермінації:"
HorizontalAlignment="Right"></Label>
        <Label Height="25" Width="86" Margin="0,40,214,1" Name="R1"
HorizontalAlignment="Right"></Label>
        <Label Height="25" Width="100" Margin="0,10,100,31" Content="Байєса-Шварца:"
HorizontalAlignment="Right"></Label>
        <Label Height="25" Width="90" Margin="0,10,10,31" Name="BS1"
HorizontalAlignment="Right"></Label>
        <Label Height="25" Width="109" Margin="0,40,100,1" Content="Дарбіна-Уотсона:"
HorizontalAlignment="Right"></Label>
        <Label Height="25" Width="90" Margin="0,40,10,1" Name="DW1"
HorizontalAlignment="Right"></Label>
    </Grid>
</GroupBox>
</Grid>
</TabItem>
<TabItem Header="Тестова">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="2*"></RowDefinition>
            <RowDefinition></RowDefinition>
        </Grid.RowDefinitions>
        <Ivc:CartesianChart Series="{ Binding SeriesCollection2}" LegendLocation="Right" >
            <Ivc:CartesianChart.AxisY>
                <Ivc:Axis LabelFormatter="{ Binding YFormatter}"></Ivc:Axis>
            </Ivc:CartesianChart.AxisY>
            <Ivc:CartesianChart.AxisX>
                <Ivc:Axis Labels="{ Binding Labels}"></Ivc:Axis>
            </Ivc:CartesianChart.AxisX>
        </Ivc:CartesianChart>
        <GroupBox x:Name="groupBox_Copy" Header="Критерії якості моделі для тестової вибірки" Height="88"
Width="600" Margin="0,0,0,0" Grid.Row="1">
            <Grid>
                <Label Height="25" Width="44" Margin="0,10,525,31" Content="RMSE:"
HorizontalAlignment="Right"></Label>
                <Label Height="25" Width="75" Margin="0,10,450,31" Name="RMSE2"
HorizontalAlignment="Right"></Label>
                <Label Height="25" Width="32" Margin="0,40,525,1" Content="IKA:"
HorizontalAlignment="Right"></Label>
                <Label Height="25" Width="75" Margin="0,40,450,1" Name="IKA2"
HorizontalAlignment="Right"></Label>
                <Label Height="25" Width="55" Margin="0,10,300,31" Content="Фішера:"
HorizontalAlignment="Right"></Label>
                <Label Height="25" Width="87" Margin="0,10,213,31" Name="F2"
HorizontalAlignment="Right"></Label>
                <Label Height="25" Width="145" Margin="0,40,300,1" Content="Коефіцієнт детермінації:"
HorizontalAlignment="Right"></Label>
                <Label Height="25" Width="86" Margin="0,40,214,1" Name="R2"
HorizontalAlignment="Right"></Label>
                <Label Height="25" Width="100" Margin="0,10,100,31" Content="Байєса-Шварца:"
HorizontalAlignment="Right"></Label>
                <Label Height="25" Width="90" Margin="0,10,10,31" Name="BS2"
HorizontalAlignment="Right"></Label>
                <Label Height="25" Width="109" Margin="0,40,100,1" Content="Дарбіна-Уотсона:"
HorizontalAlignment="Right"></Label>
                <Label Height="25" Width="90" Margin="0,40,10,1" Name="DW2"
HorizontalAlignment="Right"></Label>
            </Grid>
        </GroupBox>
    </Grid>
</TabItem>
<TabItem Header="Прогноз">
    <Ivc:CartesianChart Series="{ Binding SeriesCollection3}" LegendLocation="Right" >
        <Ivc:CartesianChart.AxisY>
            <Ivc:Axis LabelFormatter="{ Binding YFormatter}"></Ivc:Axis>
        </Ivc:CartesianChart.AxisY>
        <Ivc:CartesianChart.AxisX>
            <Ivc:Axis Labels="{ Binding Labels}"></Ivc:Axis>
        </Ivc:CartesianChart.AxisX>
    </Ivc:CartesianChart>

```



```

        </TabItem>
    </TabControl>
</Grid>
</TabItem>
</TabControl>

</DockPanel>

</Grid>
</Window>
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MathNet.Numerics.LinearAlgebra;
using MathNet.Numerics.LinearRegression;
using System.IO;

namespace RetailDSS.Business_Logic
{
    public static class Regression
    {
        public static class rand
        {
            public static Random random = new Random();

            // get random number in [0; 1)
            public static double get()
            {
                int randomNumber = random.Next(0, 1000);
                return randomNumber / 10000;
            }
        }

        #region calc
        private static double mat_exp(double[] Y, int N)
        {
            return Y.Sum() / N;
        }

        private static double disp(double[] Y, int N)
        {
            double res = 0;
            double exp = mat_exp(Y, N);
            for (int i = 0; i < N; i++)
                res += Math.Pow(Y[i] - exp, 2);
            res /= (N - 1);
            return res;
        }

        private static double SMD(double[] Y, int N)
        {
            return Math.Sqrt(disp(Y, N));
        }

        private static double assym(double[] Y, int N)
        {
            double res = 0;
            double exp = mat_exp(Y, N);
            double smd = SMD(Y, N);
            for (int i = 0; i < N; i++)
            {
                double t = Y[i] - exp;
                res += t * t * t;
            }

            res /= (N);

            double den = Math.Sqrt(disp(Y, N));
            den = den * den * den;
            return res / den;
        }
    }
}

```

```

private static double excess(double[] Y, int N)
{
    decimal res = 0;
    decimal exp = (decimal)mat_exp(Y, N);
    for (int i = 0; i < N; i++)
    {
        decimal t = (decimal)Y[i] - exp;
        res += t * t * t * t;
    }

    res /= (N);

    return (double)res / (disp(Y, N) * disp(Y, N));
}

private static double JB(double[] Y, int N, int regrcnt)
{
    double res = (N - regrcnt) / 6;
    double S = assym(Y, N);
    double K = excess(Y, N);
    res *= (S * S + (K - 3) * (K - 3) / 4);
    return res;
}

private static double correl(double[] X, double[] Y, int N)
{
    double exp_X = mat_exp(X, N);
    double exp_Y = mat_exp(Y, N);
    double res = 0;

    for (int i = 0; i < N; i++)
        res += (X[i] - exp_X) * (Y[i] - exp_Y);

    res /= SMD(X, N);
    res /= SMD(Y, N);
    res /= (N - 1);

    return res;
}

private static double autocorrel(double[] X, int N, int s)
{
    double exp_X = mat_exp(X, N);
    double res = 0;
    double smd = 0;

    for (int i = s; i < N; i++)
        res += (X[i] - exp_X) * (X[i - s] - exp_X);

    smd = SMD(X, N);
    res /= smd;
    res /= smd;
    res /= (N - 1);

    return res;
}

private static double RMSE(double[] Y, double[] y, int N)
{
    double res = 0;
    for (int i = 0; i < N; i++)
        res += Math.Pow(Y[i] - y[i], 2);
    return Math.Sqrt(res / N);
}

private static double determ(double[] Y, double[] y, int N)
{
    double res = 0;
    res = disp(y, N) / disp(Y, N);
    return res;
}

private static double IKA(double[] Y, double[] y, int N, int n_param)
{
    return N * Math.Log(RMSE(Y, y, N)) + 2 * n_param;
}

```

```

private static double Bayes_Schwarz(double[] Y, double[] y, int N, int n_param)
{
    return N * Math.Log(RMSE(Y, y, N)) + n_param * Math.Log(n_param);
}

private static double Fisher(double[] Y, double[] y, int N)
{
    double det = determ(Y, y, N);
    return det / (1 - det);
}

private static Vector<double> LS(Matrix<double> X, Vector<double> y)
{
    Vector<double> p = MultipleRegression.NormalEquations(X, y);
    return p;
}

private static double DW(double[] Y, double[] y, int N)
{
    double[] eps = new double[N];
    double res = 0;
    for (int i = 0; i < N; i++)
        eps[i] = Y[i] - y[i];

    double nom = 0;
    for (int i = 1; i < N; i++)
        nom += Math.Pow(eps[i] - eps[i - 1], 2);
    double den = 0;
    for (int i = 0; i < N; i++)
        den += Math.Pow(eps[i], 2);

    res = nom / den;
    return res;
}

private static double PAC(double[] X, int N, int s)
{
    return PHI(X, N, s, s);
}

private static double PHI(double[] X, int N, int k, int j)
{
    if (k == 1 && j == 1)
        return autocorrel(X, N, 1);
    if (k == 2 && j == 2)
        return (autocorrel(X, N, 2) - autocorrel(X, N, 1) * autocorrel(X, N, 1)) / (1 - autocorrel(X, N, 1) * autocorrel(X, N, 1));
    if (k == j)
    {
        double nom = autocorrel(X, N, k);
        double den = 1;

        for (int t = 1; t <= k - 1; t++)
            nom -= PHI(X, N, k - 1, t) * autocorrel(X, N, k - t);

        for (int t = 1; t <= k - 1; t++)
            den -= PHI(X, N, k - 1, t) * autocorrel(X, N, t);

        return nom / den;
    }
    return PHI(X, N, k - 1, j) - PHI(X, N, k, k) * PHI(X, N, k - 1, k - j);
}
#endregion

public static Tuple<Tuple<double[], double[], double[], double[], double[]>, Tuple<double, double, double, double, double,
double>, Tuple<double, double, double, double, double, double>, string> AR(int N_study, List<double> Y, int lag, double minPac, int N_forec)
{
    List<int> lags = new List<int>();
    int AR_lag_max_curr = 0;
    int AR_lag_cnt = 0;

    for (int lg = 1; lg < lag; lg++)
    {
        double corr = Math.Abs(PAC(Y.ToArray(), Y.Count, lg));
        double min_corr = Math.Abs(minPac);
        if (corr > min_corr)

```

```

    {
        AR_lag_max_curr = lg;
        AR_lag_cnt++;
    }
}

for (int i = 1; i < lag; i++)
{
    if (Math.Abs(PAC(Y.ToArray(), Y.Count, i)) > Math.Abs(minPac))
        lags.Add(i);
}
var X = Matrix<double>.Build.Dense(N_study - AR_lag_max_curr, AR_lag_cnt + 1);
for (int i = 0; i < N_study - AR_lag_max_curr; i++)
{
    X[i, 0] = 1;
    for (int j = 0; j < AR_lag_cnt; j++)
    {
        X[i, j + 1] = Y[i + AR_lag_max_curr - lags[j]];
    }
}
var yy = Vector<double>.Build.Dense(N_study - AR_lag_max_curr);
for (int i = 0; i < N_study - AR_lag_max_curr; i++)
{
    yy[i] = Y[i + AR_lag_max_curr];
}
var p = LS(X, yy);

string descr = "Y = ";
descr += p[0];
for (int j = 0; j < AR_lag_cnt; j++)
    //if (dataGridView5.Columns[regr].Visible == true)
    {
        descr += " + ";
        descr += p[j + 1].ToString();
        descr += " * ";
        // regr name
        descr += "y(k-" + lags[j] + ") ";
    }

double[] y = new double[N_study - AR_lag_max_curr];
for (int i = 0; i < N_study - AR_lag_max_curr; i++)
{
    y[i] = p[0];
    for (int j = 0; j < AR_lag_cnt; j++)
    {
        y[i] += p[j + 1] * Y[i + AR_lag_max_curr - lags[j]];
    }
}

List<Tuple<int, double>> model = new List<Tuple<int, double>>();
model.Add(new Tuple<int, double>(0, p[0]));
for (int i = 0; i < AR_lag_cnt; i++)
{
    model.Add(new Tuple<int, double>(lags[i], p[i + 1]));
}

double rmse = RMSE(Y.ToArray(), y, N_study - AR_lag_max_curr);
double det = determ(Y.ToArray(), y, N_study - AR_lag_max_curr);
double ika = IKA(Y.ToArray(), y, N_study - AR_lag_max_curr, 5);
double bs = Bayes_Shwarz(Y.ToArray(), y, N_study - AR_lag_max_curr, 5);
double f = Fisher(Y.ToArray(), y, N_study - AR_lag_max_curr);
double dw = DW(Y.ToArray(), y, N_study - AR_lag_max_curr);

var y1 = new double[Y.Count - N_study];
var Y1 = new double[Y.Count - N_study];
for (int i = 0; i < Y.Count - N_study; i++)
{
    Y1[i] = Y[i + N_study];
    y1[i] = p[0];
    for (int j = 0; j < AR_lag_cnt; j++)
    {
        y1[i] += p[j + 1] * Y[i + N_study - lags[j]];
    }
}

double rmse1 = RMSE(Y1, y1, Y.Count - N_study);

```

```

double det1 = determ(Y1, y1, Y.Count - N_study);
double ika1 = IKA(Y1, y1, Y.Count - N_study, 5);
double bs1 = Bayes_Schwarz(Y1, y1, Y.Count - N_study, 5);
double f1 = Fisher(Y1, y1, Y.Count - N_study);
double dw1 = DW(Y1, y1, Y.Count - N_study);

double[] Y_forec = new double[N_forec];
// used to store original sample and predicted values
double[] Y_extended = new double[Y.Count + N_forec];

for (int i = 0; i < Y.Count; i++)
    Y_extended[i] = Y[i];

for (int i = 0; i < N_forec; i++)
{
    Y_forec[i] = p[0];
    int k = i + Y.Count;
    for (int j = 0; j < AR_lag_cnt; j++)
        Y_forec[i] += p[j + 1] * Y_extended[k - lags[j]];
    Y_extended[Y.Count + i] = y[i];
}

//RegressionToCSV(model, rmse, det, ika, bs, f, dw, @"C:\Users\ttarasiuk\Documents\SY\Disertation\Data\Results\AR(" + lag
+ ").csv");
return new Tuple<Tuple<double[], double[], double[], double[], double[]>, Tuple<double, double, double, double, double,
double>, Tuple<double, double, double, double, double, double>, string>(new Tuple<double[], double[], double[], double[],
double[]>(Y.ToArray(), y.ToArray(), Y1, y1, Y_forec), new Tuple<double, double, double, double, double, double>(rmse, det, ika, bs, f, dw),
new Tuple<double, double, double, double, double, double>(rmse1, det1, ika1, bs1, f1, dw1), descr);
}

public static void RegressionToCSV(List<Tuple<int, double>> model, double rmse, double det, double ika, double bs, double
fisher, double dw, string path)
{
    List<string> lines = new List<string>();
    string header = "Modeled y;RMSE;Deter;IKA;Bayes_Schwarz;Fisher;DW";
    string line = "y(t) = " + model[0].Item2;
    for (int i = 1; i < model.Count; i++)
    {
        line += " + " + model[i].Item2 + "* y(t - " + model[i].Item1 + ")";
    }
    line += "; " + rmse + "; " + det + "; " + ika + "; " + bs + "; " + fisher + "; " + dw;
    lines.Add(header);
    lines.Add(line);

    File.WriteAllLines(path, lines);
}

public static void Trend(int N_study, List<double> Y, int power)
{
    var X = Matrix<double>.Build.Dense(N_study, power + 1);
    for (int k = 0; k < N_study; k++)
    {
        X[k, 0] = 1;

        for (int pow = 0; pow < power; pow++)
            X[k, pow + 1] = Convert.ToDouble(Math.Pow(k + 1, pow + 1));
    }
    var yy = Vector<double>.Build.Dense(N_study);
    for (int i = 0; i < N_study; i++)
    {
        yy[i] = Y[i];
    }
    var p = LS(X, yy);

    double[] y = new double[N_study];
    for (int i = 0; i < N_study; i++)
    {
        y[i] = p[0];
        for (int pow = 0; pow < power; pow++)
            y[i] += p[pow + 1] * Math.Pow(i + 1, pow + 1);
    }

    List<Tuple<int, double>> model = new List<Tuple<int, double>>();
    for (int i = 0; i < p.Count; i++)
    {
        model.Add(new Tuple<int, double>(i, p[i]));
    }
}

```

```

    }
    double rmse = RMSE(Y.ToArray(), y, N_study);
    double det = determ(Y.ToArray(), y, N_study);
    double ika = IKA(Y.ToArray(), y, N_study, 5);
    double bs = Bayes_Shwarz(Y.ToArray(), y, N_study, 5);
    double f = Fisher(Y.ToArray(), y, N_study);
    double dw = DW(Y.ToArray(), y, N_study);

    RegressionToCSV(model, rmse, det, ika, bs, f, dw, @"C:\Users\ttarasiuk\Documents\SY\Disertation\Data\Results\Trend(" +
power + ").csv");
}

public enum MovAv
{
    Exponential,
    Previous,
    Mean
}

public static void CalcMovAv(double[] Y, int N, double[] MA, MovAv type, int window)
{
    int k;
    int cnt;
    switch (type)
    {
        case MovAv.Exponential:
        {
            double alpha = 2 / (window + 1);
            for (k = 0; k < N; k++)
            {
                MA[k] = 0;
                cnt = 0;
                for (int i = k; i > k - window; i--)
                {
                    if (i >= 0 && i < N)
                    {
                        double koef = Math.Pow(1 - alpha, k - i);
                        MA[k] += koef * Y[i];
                        cnt++;
                    }
                }
                MA[k] /= cnt;
            }
            break;
        }
        case MovAv.Mean:
        {
            for (k = 0; k < N; k++)
            {
                MA[k] = 0;
                cnt = 0;
                for (int i = k - window / 2; i < k + window / 2; i++)
                {
                    if (i >= 0 && i < N)
                    {
                        MA[k] += Y[i];
                        cnt++;
                    }
                }
                MA[k] /= cnt;
            }
            break;
        }
        case MovAv.Previous:
        {
            for (k = 0; k < N; k++)
            {
                MA[k] = 0;
                cnt = 0;
                for (int i = k; i > k - window; i--)
                {
                    if (i >= 0 && i < N)
                    {
                        MA[k] += Y[i];
                        cnt++;
                    }
                }
                MA[k] /= cnt;
            }
            break;
        }
    }
}

```

```

    }

    public static void ARMA(int N_study, List<double> Y, int lag, int wind_size, double minPAC, double minPACforMA, MovAv
MAtype)
    {
        List<int> lags = new List<int>();
        int AR_lag_max_curr = 0;
        int AR_lag_cnt = 0;

        for (int lg = 1; lg < lag; lg++)
        {
            double corr = Math.Abs(PAC(Y.ToArray(), Y.Count, lg));
            double min_corr = Math.Abs(minPAC);
            if (corr > min_corr)
            {
                AR_lag_max_curr = lg;
                AR_lag_cnt++;
            }
        }

        for (int i = 1; i < lag; i++)
        {
            if (Math.Abs(PAC(Y.ToArray(), Y.Count, i)) > Math.Abs(minPAC))
                lags.Add(i);
        }
        var X = Matrix<double>.Build.Dense(N_study - AR_lag_max_curr, AR_lag_cnt + 1);
        for (int i = 0; i < N_study - AR_lag_max_curr; i++)
        {
            X[i, 0] = 1;
            for (int j = 0; j < AR_lag_cnt; j++)
            {
                X[i, j + 1] = Y[i + AR_lag_max_curr - lags[j]];
            }
        }
        var yy = Vector<double>.Build.Dense(N_study - AR_lag_max_curr);
        for (int i = 0; i < N_study - AR_lag_max_curr; i++)
        {
            yy[i] = Y[i + AR_lag_max_curr];
        }
        var p = LS(X, yy);

        double[] y = new double[N_study - AR_lag_max_curr];
        for (int i = 0; i < N_study - AR_lag_max_curr; i++)
        {
            y[i] = p[0];
            for (int j = 0; j < AR_lag_cnt; j++)
            {
                y[i] += p[j + 1] * Y[i + AR_lag_max_curr - lags[j]];
            }
            // mag
            double dev = (rand.get() - 0.5) * 0.02 * Convert.ToDouble(mat_exp(Y.ToArray(), Y.Count));
            y[i] += dev;
        }

        // residuals - eps
        int eps_len = N_study - AR_lag_max_curr;
        double[] eps = new double[eps_len];
        for (int i = 0; i < eps_len; i++)
            eps[i] = Y[i] - y[i];

        // build mov aver based on eps[]
        double[] MA_eps = new double[eps_len];

        CalcMovAv(eps, eps_len, MA_eps, MAtype, wind_size);

        // maximum possible lag of movav build based on eps series
        int MA_eps_lag_max = 5;
        // maximum selected lag of MA_eps
        int MA_eps_lag_max_curr = 0;
        // number of selected lags
        int MA_eps_lag_cnt = 0;
        for (int lg = 1; lg < MA_eps_lag_max; lg++)
        {
            double corr = Math.Abs(PAC(MA_eps, eps_len, lg));
            double min_corr = Math.Abs(minPACforMA);
            if (corr > min_corr)

```

```

    {
        MA_eps_lag_max_curr = lg;
        MA_eps_lag_cnt++;
    }
}

int l = 0;
// array of selected movav_eps lags
int[] MA_lags = new int[MA_eps_lag_cnt];
for (int lg = 1; lg < MA_eps_lag_max; lg++)
    if (Math.Abs(PAC(MA_eps, eps_len, lg)) > Math.Abs(minPACforMA))
    {
        MA_lags[l] = lg;
        l++;
    }

// actual LS for MA_eps
var X_eps = Matrix<double>.Build.Dense(eps_len - MA_eps_lag_max_curr, MA_eps_lag_cnt + 1);

for (int i = 0; i < eps_len - MA_eps_lag_max_curr; i++)
{
    X_eps[i, 0] = 1;
    int k = i + MA_eps_lag_max_curr;
    for (int j = 0; j < MA_eps_lag_cnt; j++)
        X_eps[i, j + 1] = Convert.ToDouble(MA_eps[k - MA_lags[j]]);
}

var yy_eps = Vector<double>.Build.Dense(eps_len - MA_eps_lag_max_curr);
for (int i = 0; i < eps_len - MA_eps_lag_max_curr; i++)
{
    int k = i + MA_eps_lag_max_curr;
    yy_eps[i] = eps[k];
}

//textBox5.Text = X.ToString() + yy.ToString();
var p_eps = LS(X_eps, yy_eps);

// modeled Y - AR part
double[] y_eps = new double[N_study - AR_lag_max_curr - MA_eps_lag_max_curr];
// original Y
double[] Y_eps = new double[N_study - AR_lag_max_curr - MA_eps_lag_max_curr];

for (int i = 0; i < N_study - AR_lag_max_curr - MA_eps_lag_max_curr; i++)
{
    int k = i + AR_lag_max_curr + MA_eps_lag_max_curr;
    Y_eps[i] = Y[k];

    y_eps[i] = p[0];
    for (int j = 0; j < AR_lag_cnt; j++)
        y_eps[i] += p[j + 1] * Y[k - lags[j]];

    y_eps[i] += p_eps[0];
    for (int j = 0; j < MA_eps_lag_cnt; j++)
        y_eps[i] += p_eps[j + 1] * MA_eps[k - AR_lag_max_curr - MA_lags[j]];
}

List<Tuple<int, double>> model = new List<Tuple<int, double>>();
model.Add(new Tuple<int, double>(0, p_eps[0]));
for (int i = 0; i < AR_lag_cnt; i++)
{
    model.Add(new Tuple<int, double>(lags[i], p_eps[i + 1]));
}

// Quality criteria for training
double rmse = RMSE(Y_eps, y_eps, N_study - AR_lag_max_curr - MA_eps_lag_max_curr);
double det = determ(Y_eps, y_eps, N_study - AR_lag_max_curr - MA_eps_lag_max_curr);
double ika = IKA(Y_eps, y_eps, N_study - AR_lag_max_curr - MA_eps_lag_max_curr, 5);
double bs = Bayes_Shwarz(Y_eps, y_eps, N_study - AR_lag_max_curr - MA_eps_lag_max_curr, 5);
double f = Fisher(Y_eps, y_eps, N_study - AR_lag_max_curr - MA_eps_lag_max_curr);
double dw = DW(Y_eps, y_eps, N_study - AR_lag_max_curr - MA_eps_lag_max_curr);

Console.WriteLine();

switch (MAtype)
{

```



```

        case MovAv.Exponential:
            RegressionToCSV(model, rmse, det, ika, bs, f, dw,
                @"C:\Users\ttarasiuk\Documents\SY\Disertation\Data\Results\ARMA(" + lag + ") Exp.csv");
            break;
        case MovAv.Previous:
            RegressionToCSV(model, rmse, det, ika, bs, f, dw,
                @"C:\Users\ttarasiuk\Documents\SY\Disertation\Data\Results\ARMA(" + lag + ") Prev.csv");
            break;
        case MovAv.Mean:
            RegressionToCSV(model, rmse, det, ika, bs, f, dw,
                @"C:\Users\ttarasiuk\Documents\SY\Disertation\Data\Results\ARMA(" + lag + ") Mean.csv");
            break;
        default:
            break;
    }
}

}

}
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace RetailDSS.Models.Business_Logic
{
    public static class Calculations
    {
        #region Read Data
        public static List<Transaction> GetTransactions(string path)
        {
            List<Transaction> transactions = File.ReadAllLines(path)
                .Skip(1)
                .Select(v => Transaction.FromCsv(v))
                //.Take(20000)
                .ToList();

            return transactions;
        }

        public static List<Item_Category> GetItemCategories(string path)
        {
            List<Item_Category> itemCategories = File.ReadAllLines(path)
                .Skip(1)
                .Select(v => Item_Category.FromCsv(v))
                .ToList();

            return itemCategories;
        }

        public static List<Product_Group> GetProductGroups(string path)
        {
            List<Product_Group> productGroups = File.ReadAllLines(path)
                .Skip(1)
                .Select(v => Product_Group.FromCsv(v))
                .ToList();

            return productGroups;
        }

        public static List<Item> GetItems(string path)
        {
            List<Item> items = File.ReadAllLines(path)
                .Skip(1)
                .Select(v => Item.FromCsv2(v))
                .ToList();

            return items;
        }

        public static List<YValue> GetY(string path)
        {
            List<YValue> Y = File.ReadAllLines(path)
                .Skip(1)
                .Select(v => YValue.GetYFromCsv(v))
                .ToList();
        }
    }
}

```

```

        return Y;
    }
#endregion

#region ABC
public static Tuple<List<Item_Category>, List<Item_Category>, List<Item_Category>, List<string>>
ABCItemCategories(List<Transaction> transactions, List<Item_Category> itemCategories, bool forProfit, double profitA, double profitB, double
profitC)
{
    List<Tuple<Item_Category, double>> Profits = new List<Tuple<Item_Category, double>>();
    foreach (var itemCategory in itemCategories)
    {
        double categoryProfit = 0;
        if (forProfit)
        {
            categoryProfit = transactions.Where(s => s.ItemCategoryCode == itemCategory.Code)
                .Sum(s => (s.Price - s.NetPrice) * s.Quantity);
        }
        else
        {
            categoryProfit = transactions.Where(s => s.ItemCategoryCode == itemCategory.Code)
                .Sum(s => s.Quantity);
        }
        Profits.Add(new Tuple<Item_Category, double>(itemCategory, categoryProfit));
    }
    List<Tuple<Item_Category, double>> SoretdProfits = Profits.OrderByDescending(o => o.Item2).ToList();

    List<Tuple<Item_Category, double>> itemCategoriesA = new List<Tuple<Item_Category, double>>();
    List<Tuple<Item_Category, double>> itemCategoriesB = new List<Tuple<Item_Category, double>>();
    List<Tuple<Item_Category, double>> itemCategoriesC = new List<Tuple<Item_Category, double>>();

    double totalSum = SoretdProfits.Sum(x => x.Item2);
    profitA *= totalSum;
    profitB *= totalSum;
    profitC *= totalSum;
    double totalA = 0;
    double totalB = 0;

    foreach (var itemCategoryProfit in SoretdProfits)
    {
        if (totalA < profitA)
        {
            itemCategoriesA.Add(itemCategoryProfit);
            totalA += itemCategoryProfit.Item2;
        }
        else
        {
            if (totalB < profitB)
            {
                itemCategoriesB.Add(itemCategoryProfit);
                totalB += itemCategoryProfit.Item2;
            }
            else
            {
                itemCategoriesC.Add(itemCategoryProfit);
            }
        }
    }

    List<string> lines = new List<string>();
    if (forProfit)
        lines.Add("Category;Code;Description;Profit");
    else
        lines.Add("Category;Code;Description;Quantity");
    foreach (var item in itemCategoriesA)
    {
        if (item.Item2 > 0)
            lines.Add("A;" + item.Item1.Code + ";" + item.Item1.Description + ";" + item.Item2);
    }
    foreach (var item in itemCategoriesB)
    {
        if (item.Item2 > 0)
            lines.Add("B;" + item.Item1.Code + ";" + item.Item1.Description + ";" + item.Item2);
    }
    foreach (var item in itemCategoriesC)
    {

```

```

        if (item.Item2 > 0)
            lines.Add("C;" + item.Item1.Code + ";" + item.Item1.Description + ";" + item.Item2);
    }

    List<Item_Category> CategoriesA = new List<Item_Category>();
    foreach (var item in itemCategoriesA)
    {
        CategoriesA.Add(item.Item1);
    }
    List<Item_Category> CategoriesB = new List<Item_Category>();
    foreach (var item in itemCategoriesB)
    {
        CategoriesB.Add(item.Item1);
    }
    List<Item_Category> CategoriesC = new List<Item_Category>();
    foreach (var item in itemCategoriesC)
    {
        CategoriesC.Add(item.Item1);
    }

    return new Tuple<List<Item_Category>, List<Item_Category>, List<Item_Category>, List<string>>(CategoriesA,
CategoriesB, CategoriesC, lines);
}

public static Tuple<List<Product_Group>, List<Product_Group>, List<Product_Group>, List<string>>
ABCProductGroups(List<Transaction> transactions, List<Product_Group> productGroups, bool forProfit, double profitA, double profitB, double
profitC)
{
    List<Tuple<Product_Group, double>> Profits = new List<Tuple<Product_Group, double>>();
    foreach (var productGroup in productGroups)
    {
        double categoryProfit = 0;
        if (forProfit)
        {
            categoryProfit = transactions.Where(s => s.ProductGroupCode == productGroup.Code)
                .Sum(s => (s.Price - s.NetPrice) * s.Quantity);
        }
        else
        {
            categoryProfit = transactions.Where(s => s.ProductGroupCode == productGroup.Code)
                .Sum(s => s.Quantity);
        }
        Profits.Add(new Tuple<Product_Group, double>(productGroup, categoryProfit));
    }
    List<Tuple<Product_Group, double>> SoretdProfits = Profits.OrderByDescending(o => o.Item2).ToList();

    List<Tuple<Product_Group, double>> productGroupsA = new List<Tuple<Product_Group, double>>();
    List<Tuple<Product_Group, double>> productGroupsB = new List<Tuple<Product_Group, double>>();
    List<Tuple<Product_Group, double>> productGroupsC = new List<Tuple<Product_Group, double>>();

    double totalSum = SoretdProfits.Sum(x => x.Item2);
    profitA *= totalSum;
    profitB *= totalSum;
    profitC *= totalSum;
    double totalA = 0;
    double totalB = 0;

    foreach (var productGroupProfit in SoretdProfits)
    {
        if (totalA < profitA)
        {
            productGroupsA.Add(productGroupProfit);
            totalA += productGroupProfit.Item2;
        }
        else
        {
            if (totalB < profitB)
            {
                productGroupsB.Add(productGroupProfit);
                totalB += productGroupProfit.Item2;
            }
            else
            {
                productGroupsC.Add(productGroupProfit);
            }
        }
    }
}

```

```

    }

    List<string> lines = new List<string>();
    if (forProfit)
        lines.Add("Category;Code;Description;Profit");
    else
        lines.Add("Category;Code;Description;Quantity");
    foreach (var item in productGroupsA)
    {
        if (item.Item2 > 0)
            lines.Add("A;" + item.Item1.Code + ";" + item.Item1.Description + ";" + item.Item2);
    }
    foreach (var item in productGroupsB)
    {
        if (item.Item2 > 0)
            lines.Add("B;" + item.Item1.Code + ";" + item.Item1.Description + ";" + item.Item2);
    }
    foreach (var item in productGroupsC)
    {
        if (item.Item2 > 0)
            lines.Add("C;" + item.Item1.Code + ";" + item.Item1.Description + ";" + item.Item2);
    }

    List<Product_Group> groupsA = new List<Product_Group>();
    foreach (var item in productGroupsA)
    {
        groupsA.Add(item.Item1);
    }
    List<Product_Group> groupsB = new List<Product_Group>();
    foreach (var item in productGroupsB)
    {
        groupsB.Add(item.Item1);
    }
    List<Product_Group> groupsC = new List<Product_Group>();
    foreach (var item in productGroupsC)
    {
        groupsC.Add(item.Item1);
    }

    return new Tuple<List<Product_Group>, List<Product_Group>, List<Product_Group>, List<string>>(groupsA, groupsB,
groupsC, lines);
}

public static Tuple<List<Item>, List<Item>, List<Item>, List<string>> ABCItems(List<Transaction> transactions, List<Item>
items, bool forProfit, double profitA, double profitB, double profitC)
{
    List<Tuple<Item, double>> Profits = new List<Tuple<Item, double>>();
    foreach (var item in items)
    {
        double categoryProfit = 0;
        if (forProfit)
        {
            categoryProfit = transactions.Where(s => s.ItemNo == item.No)
                .Sum(s => (s.Price - s.NetPrice) * s.Quantity);
        }
        else
        {
            categoryProfit = transactions.Where(s => s.ItemNo == item.No)
                .Sum(s => s.Quantity);
        }
        Profits.Add(new Tuple<Item, double>(item, categoryProfit));
    }
    List<Tuple<Item, double>> SoretdProfits = Profits.OrderByDescending(o => o.Item2).ToList();

    List<Tuple<Item, double>> ItemsA = new List<Tuple<Item, double>>();
    List<Tuple<Item, double>> ItemsB = new List<Tuple<Item, double>>();
    List<Tuple<Item, double>> ItemsC = new List<Tuple<Item, double>>();

    double totalSum = SoretdProfits.Sum(x => x.Item2);
    profitA *= totalSum;
    profitB *= totalSum;
    profitC *= totalSum;
    double totalA = 0;
    double totalB = 0;

    foreach (var itemProfit in SoretdProfits)

```

```

    {
        if (totalA < profitA)
        {
            ItemsA.Add(itemProfit);
            totalA += itemProfit.Item2;
        }
        else
        {
            if (totalB < profitB)
            {
                ItemsB.Add(itemProfit);
                totalB += itemProfit.Item2;
            }
            else
            {
                ItemsC.Add(itemProfit);
            }
        }
    }

    List<string> lines = new List<string>();
    if (forProfit)
        lines.Add("Category;No;Description;Profit");
    else
        lines.Add("Category;No;Description;Quantity");
    foreach (var item in ItemsA)
    {
        if (item.Item2 > 0)
            lines.Add("A;" + item.Item1.No + ";" + item.Item1.Description + ";" + item.Item2);
    }
    foreach (var item in ItemsB)
    {
        if (item.Item2 > 0)
            lines.Add("B;" + item.Item1.No + ";" + item.Item1.Description + ";" + item.Item2);
    }
    foreach (var item in ItemsC)
    {
        if (item.Item2 > 0)
            lines.Add("C;" + item.Item1.No + ";" + item.Item1.Description + ";" + item.Item2);
    }

    List<Item> itmsA = new List<Item>();
    foreach (var item in ItemsA)
    {
        itmsA.Add(item.Item1);
    }
    List<Item> itmsB = new List<Item>();
    foreach (var item in ItemsB)
    {
        itmsB.Add(item.Item1);
    }
    List<Item> itmsC = new List<Item>();
    foreach (var item in ItemsC)
    {
        itmsC.Add(item.Item1);
    }

    return new Tuple<List<Item>, List<Item>, List<Item>, List<string>>(itmsA, itmsB, itmsC, lines);
}

#endregion

#region Split Transactions
public static void SplitTransactionsByMonth(List<Transaction> transactions)
{
    int year;
    int month;
    for (int i = 0; i < 24; i++)
    {
        if (i < 12)
        {
            year = 2016;
            month = i + 1;
        }
        else
        {

```

```

        year = 2017;
        month = i - 11;
    }
    List<Transaction> monthlyTransactions = transactions.Where(s => (s.Date.Month == month) && (s.Date.Year ==
year)).OrderBy(x => x.Date).ToList();
    List<string> lines = new List<string>();
    lines.Add("Transaction No.;Line No.;Item No.;Item Category Code;Product Group Code;Price;Net Price;Quantity;Date;");
    foreach (var item in monthlyTransactions)
    {
        lines.Add(item.TransactionNo + ";" +
            item.LineNo + ";" +
            item.ItemNo + ";" +
            item.ItemCategoryCode + ";" +
            item.ProductGroupCode + ";" +
            item.Price + ";" +
            item.NetPrice + ";" +
            item.Quantity + ";" +
            item.Date.Date);
    }
    File.WriteAllLines(@"C:\Users\ttarasiuk\Documents\SY\Disertation\Data\Trans. Sales Entry " + month + " " + year + ".csv",
lines);
    }
}

public static void SplitTransactionsByWeek(List<Transaction> transactions)
{
    int week;
    for (int i = 0; i < 27; i++)
    {
        week = i + 1;
        List<Transaction> weeklyTransactions = transactions.Where(s => ((s.Date.DayOfYear - 1) / 7 == week - 1)).OrderBy(x =>
x.Date).ToList();
        List<string> lines = new List<string>();
        lines.Add("Transaction No.;Line No.;Item No.;Item Category Code;Product Group Code;Price;Net Price;Quantity;Date;");
        foreach (var item in weeklyTransactions)
        {
            lines.Add(item.TransactionNo + ";" +
                item.LineNo + ";" +
                item.ItemNo + ";" +
                item.ItemCategoryCode + ";" +
                item.ProductGroupCode + ";" +
                item.Price + ";" +
                item.NetPrice + ";" +
                item.Quantity + ";" +
                item.Date.Date);
        }
        File.WriteAllLines(@"C:\Users\ttarasiuk\Documents\SY\Disertation\Data\Trans. Sales Entry " + week + " week 2016.csv",
lines);
    }
}
#endregion

public static List<Tuple<Item_Category, List<Item>>>> GetItemsByCategories(List<Item_Category> categories, List<Item>
items)
{
    List<Tuple<Item_Category, List<Item>>>> itemsByCategories = new List<Tuple<Item_Category, List<Item>>>>();
    foreach (var category in categories)
    {
        var itemsByCategory = items.Select(x => x)
            .Where(x => x.ItemCategoryCode == category.Code).ToList();
        itemsByCategories.Add(new Tuple<Item_Category, List<Item>>>>(category, itemsByCategory));
    }
    return itemsByCategories;
}

public static List<Tuple<Item_Category, List<Transaction>>>> GetTransactionsByCategories(List<Item_Category> categories,
List<Transaction> transactions)
{
    List<Tuple<Item_Category, List<Transaction>>>> transactionsByCategories = new List<Tuple<Item_Category,
List<Transaction>>>>();
    foreach (var category in categories)
    {
        var transactionsByCategory = transactions.Select(x => x)
            .Where(x => x.ItemCategoryCode == category.Code).ToList();
        transactionsByCategories.Add(new Tuple<Item_Category, List<Transaction>>>>(category, transactionsByCategory));
    }
}

```

```

        return transactionsByCategories;
    }

    #region Get Regression Data
    // Retrieves <Item_Category, Quantity, Profit> list
    public static List<Tuple<Item_Category, List<double>, List<double>>> GetRegressionDataByCategories(List<Transaction>
transactions, List<Item_Category> itemCategories)
    {
        var regressionDataByCategories = new List<Tuple<Item_Category, List<double>, List<double>>>();
        int week;
        int NumberOfWeeks = 27;
        foreach (var itemCategory in itemCategories)
        {
            List<double> profits = new List<double>();
            List<double> quantities = new List<double>();
            for (int i = 0; i < NumberOfWeeks; i++)
            {
                week = i + 1;
                double weeklyQuantity = transactions.Where(s => ((s.Date.DayOfYear - 1) / 7 == week - 1) && (s.ItemCategoryCode ==
itemCategory.Code))
                    .Sum(x => x.Quantity);
                double weeklyProfit = transactions.Where(s => ((s.Date.DayOfYear - 1) / 7 == week - 1) && (s.ItemCategoryCode ==
itemCategory.Code))
                    .Sum(x => (x.Price - x.NetPrice) * x.Quantity);
                quantities.Add(weeklyQuantity);
                profits.Add(weeklyProfit);
            }
            regressionDataByCategories.Add(new Tuple<Item_Category, List<double>, List<double>>>(itemCategory, quantities,
profits));
        }

        return regressionDataByCategories;
    }

    // Retrieves <Item, Quantity, Profit> list
    public static List<Tuple<Item, List<double>, List<double>>> GetRegressionDataByItems(List<Transaction> transactions,
List<Item> items)
    {
        var regressionDataByItems = new List<Tuple<Item, List<double>, List<double>>>();
        int week;
        int NumberOfWeeks = 27;
        foreach (var item in items)
        {
            List<double> profits = new List<double>();
            List<double> quantities = new List<double>();
            for (int i = 0; i < NumberOfWeeks; i++)
            {
                week = i + 1;
                double weeklyQuantity = transactions.Where(s => ((s.Date.DayOfYear - 1) / 7 == week - 1) && (s.ItemNo == item.No))
                    .Sum(x => x.Quantity);
                double weeklyProfit = transactions.Where(s => ((s.Date.DayOfYear - 1) / 7 == week - 1) && (s.ItemNo == item.No))
                    .Sum(x => (x.Price - x.NetPrice) * x.Quantity);
                quantities.Add(weeklyQuantity);
                profits.Add(weeklyProfit);
            }
            regressionDataByItems.Add(new Tuple<Item, List<double>, List<double>>>(item, quantities, profits));
        }

        return regressionDataByItems;
    }

    #endregion

    #region Generate Regression Data
    public static void GenerateRegressionData(List<Transaction> transactions, List<Item_Category> itemCategories, bool forProfit)
    {
        int week;
        List<string> lines = new List<string>();
        string header = "Item Category";
        for (int i = 0; i < 27; i++)
        {
            header += ";Week" + (i + 1);
        }
        lines.Add(header);
        foreach (var itemCategory in itemCategories)
        {
            string data = itemCategory.Code;

```

```

        for (int i = 0; i < 27; i++)
        {
            week = i + 1;
            double weeklySum;
            if (forProfit)
            {
                weeklySum = transactions.Where(s => ((s.Date.DayOfYear - 1) / 7 == week - 1) && (s.ItemCategoryCode ==
itemCategory.Code))
                    .Sum(x => (x.Price - x.NetPrice) * x.Quantity);
            }
            else
            {
                weeklySum = transactions.Where(s => ((s.Date.DayOfYear - 1) / 7 == week - 1) && (s.ItemCategoryCode ==
itemCategory.Code))
                    .Sum(x => x.Quantity);
            }
            data += ";" + weeklySum;

        }
        lines.Add(data);
    }
    string filePath = @"C:\Users\ttarasiuk\Documents\SY\Disertation\Data\Trans. Sales Entry Weekly";
    if (forProfit)
        filePath += " Profits by Category.csv";
    else
        filePath += " Quantities by Category.csv";
    File.WriteAllLines(filePath, lines);
}

public static void GenerateRegressionDataPerDay(List<Transaction> transactions, List<Item_Category> itemCategories, bool
forProfit)
{
    int day;
    List<string> lines = new List<string>();
    string header = "Item Category";
    for (int i = 0; i < 189; i++)
    {
        header += ";Day" + (i + 1);
    }
    lines.Add(header);
    foreach (var itemCategory in itemCategories)
    {
        if (transactions.Where(x => x.ItemCategoryCode == itemCategory.Code).Sum(x => x.Quantity) > 0)
        {
            string data = itemCategory.Code;
            for (int i = 0; i < 189; i++)
            {
                day = i + 1;
                double daylySum;
                if (forProfit)
                {
                    daylySum = transactions.Where(s => (s.Date.DayOfYear == day) && (s.ItemCategoryCode == itemCategory.Code))
                        .Sum(x => (x.Price - x.NetPrice) * x.Quantity);
                }
                else
                {
                    daylySum = transactions.Where(s => (s.Date.DayOfYear == day) && (s.ItemCategoryCode == itemCategory.Code))
                        .Sum(x => x.Quantity);
                }
                data += ";" + daylySum;
            }
            lines.Add(data);
        }
    }
    string filePath = @"C:\Users\ttarasiuk\Documents\SY\Disertation\Data\Dayly\Categories Total\Trans. Sales Entry Weekly ";
    if (forProfit)
        filePath += "Profits by Category.csv";
    else
        filePath += "Quantities by Category.csv";
    File.WriteAllLines(filePath, lines);
}

public static void GenerateRegressionDataTransactionsPerDay(List<Transaction> transactions)
{
    int day;

```



```

List<string> lines = new List<string>();
string header = "Transactions";
for (int i = 0; i < 189; i++)
{
    header += ";Day" + (i + 1);
}
lines.Add(header);
string data = "";
for (int i = 0; i < 189; i++)
{
    day = i + 1;
    double dailyCount;
    dailyCount = transactions.Where(s => (s.Date.DayOfYear == day))
        .DistinctBy(x => x.TransactionNo)
        .Count();
    data += ";" + dailyCount;
}
lines.Add(data);
string filePath = @"C:\Users\ttarasiuk\Documents\SY\Disertation\Data\Dayly\Transactions Total\Trans. Sales Entry
Dayly.csv";
File.WriteAllLines(filePath, lines);
}
public static IEnumerable<TSource> DistinctBy<TSource, TKey>(this IEnumerable<TSource> source, Func<TSource, TKey>
keySelector)
{
    HashSet<TKey> seenKeys = new HashSet<TKey>();
    foreach (TSource element in source)
    {
        if (seenKeys.Add(keySelector(element)))
        {
            yield return element;
        }
    }
}
#endregion
}
}

using RetailDSS.Models;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace RetailDSS.Business_Logic
{
    public static class Apriori
    {
        #region Init
        private static List<Item_Category> InitItemCategories()
        {
            List<Item_Category> itemCategories = new List<Item_Category>()
            {
                new Item_Category("962", "Restaurant."),
                new Item_Category("151", "Dining and serving"),
                new Item_Category("963", "Bistro"),
                new Item_Category("163", "Home decoration"),
                new Item_Category("152", "Glassware"),
                new Item_Category("182", "Clothes and shoes organisation"),
                new Item_Category("181", "Small storage."),
                new Item_Category("143", "Storing and washing"),
                new Item_Category("142", "Kitchen tools"),
                new Item_Category("961", "Swedish Food Market"),
                new Item_Category("162", "Green decoration"),
                new Item_Category("113", "Bath textiles"),
                new Item_Category("076", "Kitchen accessories"),
                new Item_Category("155", "Coffee and tea."),
                new Item_Category("141", "Cookware"),
                new Item_Category("062", "Bathroom organisers"),
                new Item_Category("123", "Cushions, throws and chairpads"),
                new Item_Category("941", "Administrative Range 941"),
                new Item_Category("021", "Multipurpose furniture"),
                new Item_Category("164", "Wall decoration."),
                new Item_Category("111", "Bedlinen"),
            }
        }
    }
}

```

```

        new Item_Category("154", "Set the table"),
        new Item_Category("091", "Baby"),
        new Item_Category("121", "Window solutions."),
        new Item_Category("094", "Play")
    };

    return itemCategories;
}
#endregion

#region Output

private static void AprioriItemCategoriesToCSV(List<Tuple<HashSet<Item_Category>, HashSet<Item_Category>, double,
double, double>> set, string path)
{
    List<string> lines = new List<string>();
    string header = "Rules;Support;Confidence;Improvement";
    lines.Add(header);
    foreach (var item in set)
    {
        string line = "{ ";
        foreach (var item1 in item.Item1)
        {
            line += item1.Code + " ";
        }
        line += " } -> { ";
        foreach (var item1 in item.Item2)
        {
            line += item1.Code + " ";
        }
        line += " }";
        line += item.Item3.ToString("F4") + ";" + item.Item4.ToString("F4") + ";" + item.Item5.ToString("F4") + ";";
        lines.Add(line);
    }

    File.WriteAllLines(path, lines);
}

private static void AprioriItemsToCSV(List<Tuple<HashSet<Item>, HashSet<Item>, double, double, double>> set, string path)
{
    List<string> lines = new List<string>();
    string header = "Rules;Support;Confidence;Improvement";
    lines.Add(header);
    foreach (var item in set)
    {
        string line = "{ ";
        foreach (var item1 in item.Item1)
        {
            line += item1.No + " ";
        }
        line += " } -> { ";
        foreach (var item1 in item.Item2)
        {
            line += item1.No + " ";
        }
        line += " }";
        line += item.Item3.ToString("F4") + ";" + item.Item4.ToString("F4") + ";" + item.Item5.ToString("F4") + ";";
        lines.Add(line);
    }

    File.WriteAllLines(path, lines);
}

#endregion
static readonly string APRIORY_PATH = @"C:\Users\ttarasiuk\Documents\SY\Disertation\Data\Results\Apriori\";
private static double Support(int setNumber, int totalNumber)
{
    return (double) setNumber / totalNumber;
}

private static double ConfidenceX(int setX, int setXY, int totalNumber)
{
    return Support(setXY, totalNumber) / Support(setX, totalNumber);
}

private static double Improvement(int setX, int setY, int setXY, int totalNumber)

```

```

    {
        return Support(setXY, totalNumber) / (Support(setX, totalNumber) * Support(setY, totalNumber)); // should be > 1
    }

private static int CountEqualElements(List<HashSet<int>> transactionsSets)
{
    List<HashSet<int>> sets = new List<HashSet<int>>();
    foreach (var item in transactionsSets)
    {
        HashSet<int> set = new HashSet<int>();
        foreach (var item1 in item)
        {
            set.Add(item1);
        }
        sets.Add(set);
    }
    for (int i = 1; i < sets.Count; i++)
    {
        sets[0].IntersectWith(sets[i]);
    }

    return sets[0].Count;
}

// Support, Confidence, Improvement
public static List<Tuple<HashSet<T>, HashSet<T>, double, double, double>> CalculateConfidence<T>(List<Tuple<T,
HashSet<int>>> itemTransactionSets, List<Tuple<HashSet<T>, int>> set, int totalTransactions, double minSupport, double minConfidence)
{
    var result = new List<Tuple<HashSet<T>, HashSet<T>, double, double, double>>();
    var set1 = new HashSet<T>();
    var set2 = new HashSet<T>();
    List<List<Tuple<HashSet<T>, HashSet<T>>>> combinations = new List<List<Tuple<HashSet<T>, HashSet<T>>>>();
    foreach (var item in set)
    {
        combinations.Add(GetAllPossibleCombitions(item.Item1));
    }
    foreach (var items in combinations)
    {
        foreach (var item in items)
        {
            HashSet<T> setXY = new HashSet<T>(item.Item1);
            setXY.UnionWith(item.Item2);
            int linkedTransactionsSetXY = CountLinkedTransactions(itemTransactionSets, setXY);
            double support = Support(linkedTransactionsSetXY, totalTransactions);
            double confidence = ConfidenceX(CountLinkedTransactions(itemTransactionSets, item.Item1),
linkedTransactionsSetXY, totalTransactions);
            if (support > minSupport && confidence > minConfidence)
                result.Add(new Tuple<HashSet<T>, HashSet<T>, double, double, double>(item.Item1,
item.Item2,
support,
confidence,
Improvement(CountLinkedTransactions(itemTransactionSets, item.Item1),
CountLinkedTransactions(itemTransactionSets, item.Item2),
linkedTransactionsSetXY,
totalTransactions))));
        }
    }
    return result;
}

public static int CountTotalTransaction(List<Transaction> transactions)
{
    return transactions.Select(x => x.TransactionNo).Distinct().Count();
}

private static int CountTotalTransactionByCategory(List<Transaction> transactions, Item_Category itemCategory)
{
    var categoryTransactions = transactions.Select(x => x).Where(x => x.ItemCategoryCode == itemCategory.Code).ToList();
    int totalTransactionsByCategory = categoryTransactions.Select(x => x.TransactionNo).Distinct().Count();
    return totalTransactionsByCategory;
}

private static HashSet<int> GetLinkedTransactions<T>(List<Tuple<T, HashSet<int>>> itemTransactionSets, HashSet<T> set)
{
    List<Tuple<T, HashSet<int>>> items = new List<Tuple<T, HashSet<int>>>();
    foreach (var item in set)

```

```

        {
            var itemTrans = itemTransactionSets.Select(x => x).Where(x => x.Item1.Equals(item)).ToList();
            items.Add(new Tuple<T, HashSet<int>>(item, itemTrans[0].Item2));
        }
        HashSet<int> setOfTrans = new HashSet<int>(items[0].Item2);
        for (int i = 1; i < items.Count; i++)
        {
            setOfTrans.IntersectWith(items[i].Item2);
        }
        return setOfTrans;
    }

    private static int CountLinkedTransactions<T>(List<Tuple<T, HashSet<int>>> itemTransactionSets, HashSet<T> set)
    {
        return GetLinkedTransactions(itemTransactionSets, set).Count;
    }

    private static List<Tuple<HashSet<T>, HashSet<T>>> GetAllPossibleCombitaions<T>(HashSet<T> set)
    {
        var possibleCombinations = new List<Tuple<HashSet<T>, HashSet<T>>>();
        var setList = set.ToList();
        var smallSet = new HashSet<T>();
        for (int i = 0; i < setList.Count; i++)
        {
            smallSet.Add(setList[i]);
            var newSet = Except(set, smallSet);
            possibleCombinations.Add(new Tuple<HashSet<T>, HashSet<T>>(new HashSet<T>(smallSet), newSet));
            AddCombination(possibleCombinations, set, setList, smallSet, i);
            smallSet.Remove(setList[i]);
        }
        return possibleCombinations;
    }

    private static HashSet<T> Except<T>(HashSet<T> setFrom, HashSet<T> set)
    {
        var resultSet = new HashSet<T>(setFrom);
        foreach (var item in set)
        {
            resultSet.Remove(item);
        }
        return resultSet;
    }

    private static void AddCombination<T>(List<Tuple<HashSet<T>, HashSet<T>>> possibleCombinations, HashSet<T> set,
    List<T> setList, HashSet<T> smallSet, int num)
    {
        for (int p = num + 1; p < setList.Count; p++)
        {
            if (smallSet.Count < setList.Count - 1)
            {
                smallSet.Add(setList[p]);
                var newSet = Except(set, smallSet);
                possibleCombinations.Add(new Tuple<HashSet<T>, HashSet<T>>(new HashSet<T>(smallSet), newSet));
                if (smallSet.Count < setList.Count - 1)
                {
                    AddCombination(possibleCombinations, set, setList, smallSet, p);
                }
                smallSet.Remove(setList[p]);
            }
        }
    }

    public static void GenerateSets<T>(List<T> itemT, List<Tuple<T, HashSet<int>>> itemTransactionSets,
    List<Tuple<HashSet<T>, int>> set2, List<Tuple<HashSet<T>, int>> set3, List<Tuple<HashSet<T>, int>> set4, List<Tuple<HashSet<T>, int>>
    set5)
    {
        for (int i = 0; i < itemTransactionSets.Count - 1; i++)
        {
            for (int j = i + 1; j < itemTransactionSets.Count; j++)
            {
                HashSet<T> itemTSet = new HashSet<T>();
                itemTSet.Add(itemT[i]);
                itemTSet.Add(itemT[j]);
                set2.Add(new Tuple<HashSet<T>, int>(itemTSet, CountEqualElements(new List<HashSet<int>>() {
                itemTransactionSets[i].Item2, itemTransactionSets[j].Item2 })));
            }
        }
    }

```

```

    }

    for (int i = 0; i < itemTransactionSets.Count - 2; i++)
    {
        for (int j = i + 1; j < itemTransactionSets.Count - 1; j++)
        {
            for (int k = j + 1; k < itemTransactionSets.Count; k++)
            {
                bool isZero = false;
                HashSet<T> itemTSet = new HashSet<T>();
                itemTSet.Add(itemT[i]);
                itemTSet.Add(itemT[j]);
                itemTSet.Add(itemT[k]);
                foreach (var item in set2)
                {
                    if (item.Item1.IsSubsetOf(itemTSet) && (item.Item2 == 0))
                        isZero = true;
                }
                if (!isZero)
                    set3.Add(new Tuple<HashSet<T>, int>(itemTSet, CountEqualElements(new List<HashSet<int>>() {
itemTransactionSets[i].Item2, itemTransactionSets[j].Item2, itemTransactionSets[k].Item2 })));
            }
        }
    }

    for (int i = 0; i < itemTransactionSets.Count - 3; i++)
    {
        for (int j = i + 1; j < itemTransactionSets.Count - 2; j++)
        {
            for (int k = j + 1; k < itemTransactionSets.Count - 1; k++)
            {
                for (int p = k + 1; p < itemTransactionSets.Count; p++)
                {
                    bool isZero = false;
                    HashSet<T> itemTSet = new HashSet<T>();
                    itemTSet.Add(itemT[i]);
                    itemTSet.Add(itemT[j]);
                    itemTSet.Add(itemT[k]);
                    itemTSet.Add(itemT[p]);
                    foreach (var item in set3)
                    {
                        if (item.Item1.IsSubsetOf(itemTSet) && (item.Item2 == 0))
                            isZero = true;
                    }
                    if (!isZero)
                        set4.Add(new Tuple<HashSet<T>, int>(itemTSet, CountEqualElements(new List<HashSet<int>>() {
itemTransactionSets[i].Item2, itemTransactionSets[j].Item2, itemTransactionSets[k].Item2, itemTransactionSets[p].Item2 })));
                }
            }
        }
    }

    //for (int i = 0; i < itemTransactionSets.Count - 4; i++)
    //{
    //    for (int j = i + 1; j < itemTransactionSets.Count - 3; j++)
    //    {
    //        for (int k = j + 1; k < itemTransactionSets.Count - 2; k++)
    //        {
    //            for (int p = k + 1; p < itemTransactionSets.Count - 1; p++)
    //            {
    //                for (int q = p + 1; q < itemTransactionSets.Count; q++)
    //                {
    //                    bool isZero = false;
    //                    HashSet<T> itemTSet = new HashSet<T>();
    //                    itemTSet.Add(itemT[i]);
    //                    itemTSet.Add(itemT[j]);
    //                    itemTSet.Add(itemT[k]);
    //                    itemTSet.Add(itemT[p]);
    //                    itemTSet.Add(itemT[q]);
    //                    foreach (var item in set4)
    //                    {
    //                        if (item.Item1.IsSubsetOf(itemTSet) && (item.Item2 == 0))
    //                            isZero = true;
    //                    }
    //                    if (!isZero)

```

```

        //          set5.Add(new Tuple<HashSet<T>, int>(itemTSet, CountEqualElements(new List<HashSet<int>>() {
itemTransactionSets[i].Item2,      itemTransactionSets[j].Item2,      itemTransactionSets[k].Item2,      itemTransactionSets[p].Item2,
itemTransactionSets[q].Item2 })));
        //          }
        //      }
        //  }
    //}
}

public static Tuple<List<Tuple<HashSet<Item_Category>, HashSet<Item_Category>, double, double, double>>,
    List<Tuple<HashSet<Item_Category>, HashSet<Item_Category>, double, double, double>>,
    List<Tuple<HashSet<Item_Category>, HashSet<Item_Category>, double, double, double>>>
RunForItemCategories(List<Transaction> transactions, List<Item_Category> allItemCategories, double minSupport, double minConfidence)
{
    List<Item_Category> itemCategories = new List<Item_Category>();
    allItemCategories = InitItemCategories();
    List<Tuple<HashSet<Item_Category>, int>> itemCategorySetsOne = new List<Tuple<HashSet<Item_Category>, int>>();
    List<Tuple<HashSet<Item_Category>, int>> itemCategorySetsTwo = new List<Tuple<HashSet<Item_Category>, int>>();
    List<Tuple<HashSet<Item_Category>, int>> itemCategorySetsThree = new List<Tuple<HashSet<Item_Category>, int>>();
    List<Tuple<HashSet<Item_Category>, int>> itemCategorySetsFour = new List<Tuple<HashSet<Item_Category>, int>>();
    List<Tuple<HashSet<Item_Category>, int>> itemCategorySetsFive = new List<Tuple<HashSet<Item_Category>, int>>();
    int totalTransactions = CountTotalTransaction(transactions);
    foreach (var item in allItemCategories)
    {
        int numberOfTransactions = transactions.Where(x => x.ItemCategoryCode == item.Code).Count();
        if (Support(numberOfTransactions, totalTransactions) > minSupport)
        {
            itemCategories.Add(item);
        }
    }

    List<Tuple<Item_Category, HashSet<int>>> itemTransactionSets = new List<Tuple<Item_Category, HashSet<int>>>();
    foreach (var item in itemCategories)
    {
        HashSet<int> itemTransactionSet = new HashSet<int>();
        var itemCategoryTransactions = transactions.Select(x => x).Where(x => x.ItemCategoryCode == item.Code).ToList();
        foreach (var trans in itemCategoryTransactions)
        {
            itemTransactionSet.Add(trans.TransactionNo);
        }
        itemTransactionSets.Add(new Tuple<Item_Category, HashSet<int>>(item, itemTransactionSet));
    }

    GenerateSets(itemCategories, itemTransactionSets, itemCategorySetsTwo, itemCategorySetsThree, itemCategorySetsFour,
itemCategorySetsFive);

    var resTwo = CalculateConfidence(itemTransactionSets, itemCategorySetsTwo, totalTransactions, minSupport,
minConfidence);
    var resThree = CalculateConfidence(itemTransactionSets, itemCategorySetsThree, totalTransactions, minSupport,
minConfidence);
    var resFour = CalculateConfidence(itemTransactionSets, itemCategorySetsFour, totalTransactions, minSupport,
minConfidence);
    return new Tuple<List<Tuple<HashSet<Item_Category>, HashSet<Item_Category>, double, double, double>>,
        List<Tuple<HashSet<Item_Category>, HashSet<Item_Category>, double, double, double>>,
        List<Tuple<HashSet<Item_Category>, HashSet<Item_Category>, double, double, double>>>(resTwo, resThree,
resFour);
}

public static List<ARule> GetCategoryRule(List<Tuple<HashSet<Item_Category>, HashSet<Item_Category>, double, double,
double>> aRule)
{
    List<ARule> rules = new List<ARule>();
    foreach (var item in aRule)
    {
        string line = "{ ";
        foreach (var item1 in item.Item1)
        {
            line += item1.Code + " ";
        }
        line += " } -> { ";
        foreach (var item1 in item.Item2)
        {
            line += item1.Code + " ";
        }
        line += " }";
    }
}

```

```

        rules.Add(new ARule(line, item.Item3, item.Item4, item.Item5));
    }
    return rules;
}

public static Tuple<List<ARule>, List<ARule>, List<ARule>> GetRulesForCategories(List<Transaction> transactions,
List<Item_Category> itemCategories, double minSupport, double minConfidence)
{
    var rules = RunForItemCategories(transactions, itemCategories, minSupport, minConfidence);
    return new Tuple<List<ARule>, List<ARule>, List<ARule>>(GetCategoryRule(rules.Item1), GetCategoryRule(rules.Item2),
GetCategoryRule(rules.Item3));
}

public static List<ARule> GetItemRule(List<Tuple<HashSet<Item>, HashSet<Item>, double, double, double>> aRule)
{
    List<ARule> rules = new List<ARule>();
    foreach (var item in aRule)
    {
        string line = "{ ";
        foreach (var item1 in item.Item1)
        {
            line += item1.No + " ";
        }
        line += " } -> { ";
        foreach (var item1 in item.Item2)
        {
            line += item1.No + " ";
        }
        line += " }";
        rules.Add(new ARule(line, item.Item3, item.Item4, item.Item5));
    }
    return rules;
}

public static Tuple<List<ARule>, List<ARule>, List<ARule>> GetRulesForItems(List<Transaction> transactions, List<Item>
items, double minSupport, double minConfidence, Item_Category itemCategory)
{
    var rules = RunForItems(transactions, items, minSupport, minConfidence, itemCategory);
    return new Tuple<List<ARule>, List<ARule>, List<ARule>>(GetItemRule(rules.Item1), GetItemRule(rules.Item2),
GetItemRule(rules.Item3));
}

public static Tuple<List<Tuple<HashSet<Item>, HashSet<Item>, double, double, double>>,
List<Tuple<HashSet<Item>, HashSet<Item>, double, double, double>>,
List<Tuple<HashSet<Item>, HashSet<Item>, double, double, double>>> RunForItems(List<Transaction>
transactions, List<Item> allItems, double minSupport, double minConfidence, Item_Category itemCategory)
{
    List<Item> items = new List<Item>();
    List<Tuple<HashSet<Item>, int>> itemSetsOne = new List<Tuple<HashSet<Item>, int>>();
    List<Tuple<HashSet<Item>, int>> itemSetsTwo = new List<Tuple<HashSet<Item>, int>>();
    List<Tuple<HashSet<Item>, int>> itemSetsThree = new List<Tuple<HashSet<Item>, int>>();
    List<Tuple<HashSet<Item>, int>> itemSetsFour = new List<Tuple<HashSet<Item>, int>>();
    List<Tuple<HashSet<Item>, int>> itemSetsFive = new List<Tuple<HashSet<Item>, int>>();
    int totalTransactionsByCategory = CountTotalTransactionByCategory(transactions, itemCategory);

    foreach (var item in allItems)
    {
        int numberOfTransactions = transactions.Where(x => x.ItemNo == item.No).Count();
        if (Support(numberOfTransactions, totalTransactionsByCategory) > minSupport)
        {
            items.Add(item);
        }
    }

    List<Tuple<Item, HashSet<int>>> itemTransactionSets = new List<Tuple<Item, HashSet<int>>>();
    foreach (var item in items)
    {
        HashSet<int> itemTransactionSet = new HashSet<int>();
        var itemTransactions = transactions.Select(x => x).Where(x => x.ItemNo == item.No).ToList();
        foreach (var trans in itemTransactions)
        {
            itemTransactionSet.Add(trans.TransactionNo);
        }
        itemTransactionSets.Add(new Tuple<Item, HashSet<int>>(item, itemTransactionSet));
    }
}

```

```

GenerateSets(items, itemTransactionSets, itemSetsTwo, itemSetsThree, itemSetsFour, itemSetsFive);

var resTwo = CalculateConfidence(itemTransactionSets, itemSetsTwo, totalTransactionsByCategory, minSupport,
minConfidence);
var resThree = CalculateConfidence(itemTransactionSets, itemSetsThree, totalTransactionsByCategory, minSupport,
minConfidence);
var resFour = CalculateConfidence(itemTransactionSets, itemSetsFour, totalTransactionsByCategory, minSupport,
minConfidence);
//var resFive = CalculateConfidence(itemTransactionSets, itemSetsFive, totalTransactionsByCategory, 0);

AprioriItemsToCSV(resTwo, APRIORY_PATH + "Item2 for Category " + itemCategory.Code + ".csv");
AprioriItemsToCSV(resThree, APRIORY_PATH + "Item3 for Category " + itemCategory.Code + ".csv");
AprioriItemsToCSV(resFour, APRIORY_PATH + "Item4 for Category " + itemCategory.Code + ".csv");
//AprioriItemsToCSV(resFive, APRIORY_PATH + "Item5 for Category " + itemCategory.Code + ".csv");
return new Tuple<List<Tuple<HashSet<Item>, HashSet<Item>, double, double, double>>, List<Tuple<HashSet<Item>,
HashSet<Item>, double, double, double>>>, List<Tuple<HashSet<Item>, HashSet<Item>, double, double, double>>>>(resTwo, resThree, resFour);
}

}
}

```